



# Database Management System (DBMS)



## LEARNING OBJECTIVES

After going through this unit, you will be able :

- ◆ to differentiate between data and information
- ◆ to know about database, the various types of databases, and why they are valuable assets for decision making
- ◆ to know the importance of database design
- ◆ to know how modern databases evolved from file systems
- ◆ to learn about flaws in file system data management
- ◆ to know the main components of the database system
- ◆ to know the main functions of a database management system (DBMS)
- ◆ to learn about the concepts of normalization in database
- ◆ to know about the database security

## Introduce Database Management System (DBMS) with its Aspects

### Data

Data are the raw (unprocessed) or isolated facts. The word raw indicates that the facts have not yet been processed to reveal their meaning. Raw data needs to be corrected to remove data entry errors. Example, Ram, 9854725697, 201, BC and Narayangadh etc. are collected facts from the user. Data itself does not have meaning. In general, data is collected and analyzed; data is only useful only after it has been analyzed in some way. Data is processed to produce meaningful and useful information. Generally data are classified as:

1. **Primary data:** Facts and figures newly collected. Examples includes: observation data, questionnaire data, surveys data, etc.
2. **Secondary data:** Facts and figures already collected. Examples includes: financial statements, customer lists, sales report, census report etc.

### Information

Information is the result of processing raw data to reveal its meaning. Data processing can be as simple as organizing data to reveal patterns or as complex as making forecasts or drawing inferences using statistical modeling. To reveal meaning, information requires context. Information provides answers to "who", "what", "where", and "when" questions. Information assigns meaning and improves the reliability of data. Example, arranging the data that we have discussed:

Name	Address	Number	Room No
Ram	Narayangadh	9854725697	BC 201

From table, we can get the information/meaning of a person of above collected data. Data and information are closely related concepts but each has its own role and has its own meaning. Sometimes data and information is taken as a same term.

### Database

A database is a shared, integrated computer structure that stores a collection of end user data, i.e. raw facts of interest to end user and metadata (data about data), through which the end user data are integrated and managed.

Database is a collection of data or information that is organized in such a way that it can be easily managed, accessed and updated. Data are stored generally in the tabular format in a database. A large quantity of data can be easily managed using any database management system. A database is built and managed using a database language called SQL (Structured Query Language). Almost all organizations have their database to organize their data in proper way. Typical commercial database applications include flight reservations, inventory management, accounting system, medical records in hospitals, records of insurance companies etc. The largest databases are usually maintained by governmental agencies, business organizations, and universities.

## Database Management System (DBMS)

A database management system (DBMS) is a collection of programs that manages the database structure and controls access to the data stored in the database. The DBMS serves as the intermediary between the user and the database. The database structure itself is stored as a collection of files, and the only way to access the data in those files is through the DBMS. DBMS presents the end user (or application program) with a single, integrated view of the data in the database. DBMS hides much of the database's internal complexity from the application programs and users. Some examples of DBMS software are MySQL, Microsoft Access, PostgreSQL, SQL Server, FileMaker, Oracle, RDBMS, dBASE, FoxPro, etc.

### Advantages of using DBMS

DBMS helps to make data management more efficient and effective. In particular, a DBMS provides advantages such as:

- **Improved data sharing:** The DBMS helps create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.
- **Improved data security:** The more users access the data, the greater the risks of data security breaches. A DBMS provides a framework for better enforcement of data privacy and security policies.
- **Better data integration:** Data integration involves combining data residing in different sources and providing users with a unified view of them. Wider access to well-managed data promotes an integrated view of the organization's operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the company affect other segments.
- **Minimized data inconsistency:** Data inconsistency occurs when different versions of the same data appear in different places. The probability of data inconsistency is greatly reduced in a properly designed database.
- **Improved data access:** DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. The DBMS makes it possible to quickly access data through queries (specific request issued to the DBMS for data manipulation). End users can get quick access of data through queries.
- **Improved decision making:** Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based. The quality of the information generated depends on the quality of the underlying data. Data quality is a comprehensive approach to promoting the accuracy, validity, and timeliness of the data. While the DBMS does not guarantee data quality, it provides a framework to facilitate data quality initiatives.

- **Increased end-user productivity:** The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure of decisions.
- **Provide Backup and Recovery:** A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery. The recovery subsystem ensures that the program is resumed from the point at which it was interrupted so that its full effect is recorded in the database.

### Disadvantages

- **Complexity of Backup and Recovery:** For a centralized shared database to be accurate and available all times, a comprehensive procedure is required to be developed and used for providing backup copies of data and for restoring a database when damage occurs.
- **Increased installation and management cost:** The large and complex DBMS software has a high initial cost. It requires trained manpower to install and operate and also has substantial annual maintenance and support cost. Additional database software may be needed to provide security and to ensure proper concurrent updating of shared data.
- **Additional hardware and software cost:** The cost of DBMS installation varies significantly, depending on the environment and functionality, size of the hardware and the recurring annual maintenance cost of hardware and software.
- **Requirement of new and specialized manpower:** Because of rapid changes in database technology and organization's business needs, the organization's need to hire, retrain its manpower on regular basis to design and implement databases, provide database administration services and manage a staff of new people. Therefore, an organization needs to maintain specialized skilled manpower.
- **Increased complexity:** A multi-user DBMS becomes an extremely complex piece of software due to expected functionality from it. It becomes necessary for database designers, developers, database administrators and end-users to understand this functionality to full advantage of it.

## Data Definition Language (DDL) and Data Manipulation Language(DML)

A database system provides a data-definition language to specify the database schema and a data manipulation language to express database queries and updates. In practice, the data-definition and data-manipulation languages are not two separate languages; instead they simply form parts of a single database language such as the widely used SQL language.

### Structured Query Language (SQL)

Structured Query Language (SQL) is a language that provides an interface to relational database systems. SQL encompasses:

- Data Manipulation Languages (DML) for INSERTs, UPDATEs, DELETEs
- Data Definition Languages (DDL) used for creating and modifying tables and other database structures.

### Features

- SQL can be used by a range of users, including those with little or no programming experience.
- It is a non procedural language.
- It reduces the amount of time required for creating and maintaining systems.
- It is an English-Like Language.

### Components of SQL

- CREATE: To create objects in the database
- ALTER: Alters the structure of the database.
- DROP: Delete objects from the database.
- TRUNCATE: Remove all records from a table, including all spaces allocated for the records are removed.
- COMMENT: Add comments to the data dictionary.
- INSERT: Insert data into a table.
- UPDATE: Updates existing data within a table.
- DELETE: Deletes all records from a table, the space for the records remain.
- LOCK TABLE: Control concurrency.

### Types of SQL Commands

#### 1. Create table command:

*Syntax:* CREATE TABLE *table name*;

Example: CREATE TABLE Employee (E-ID number (6), ENAME char (15), ADDRESS varchar (15), CITY char (15), STATE char (15));

#### 2. Insertion of Data into Tables

*Syntax:* INSERT INTO *table name*;

**Example:** INSERT INTO Employee (E-ID, ENAME, ADDRESS, CITY, STATE) VALUES (140,'TEZ','Bharatpur','Chitwan','Bagmati');

#### 3. Select Command

- To view global table data:

*Syntax:* SELECT \* FROM *table name*;

Example: SELECT \* FROM Employee;

- Retrieve ID, name, city of the employee

Example: SELECT E-ID, ENAME, CITY FROM Employee;

#### 4. Sorting data in a table

Syntax: SELECT \* FROM table name ORDER BY Column\_name1, Column\_name2 [Sort order];

Example: SELECT \* FROM Employee ORDER BY ENAME;

#### 5. Delete Operations

- Removal of All Rows

*Syntax: DELETE FROM table name;*

Example: DELETE FROM Students;

- Removal of Specified Rows

*Syntax: DELETE FROM table name WHERE search condition;*

Example: DELETE FROM Employee where EID>140;

- Update Command

Updating all Rows

*Syntax: UPDATE table name*

*SET column name = expression,*

*column name = expression;*

Example: UPDATE Employee

SET NetSal = Net\_Sal + Basic\_Sal \* 0.10;

- Updating Records Conditionally

Syntax: UPDATE table name

SET column name = expression,

column name = expression

WHERE column name = expression;

Example: UPDATE Employee

SET EName= 'Krishna',

Address = 'Kathmandu'

WHERE EID = 132;

### Data Definition Language (DDL)

The language that allows a database administrator to define the database structure, schema, and subschema is called Data definition Language. It refers to the set of SQL commands that can create and manipulate the structure of database. DDL statements are used to create, change and remove objects of a database. It enables the database administrator to define the schema components. The common DDL statements include:

**CREATE** : Generates a new table

**ALTER** : Alter or change table structure

**DROP** : Removes the table from database

**RENAME** : Rename the existing table

## Data Manipulation Language (DML)

DML is the language that allows an end user to manipulate the data in the database. It defines the environment in which data can be managed and to work with the data in the database. The common DDL statement includes:

- SELECT** : Select data from table
- INSERT** : Insert data into the table
- UPDATE** : Update existing data within a table
- DELETE** : Delete records from table

There are basically two types of DML: Procedural DMLs require a user to specify what data are needed and how to get those data and Declarative DMLs or nonprocedural DMLs require a user to specify what data are needed without specifying how to get those data.

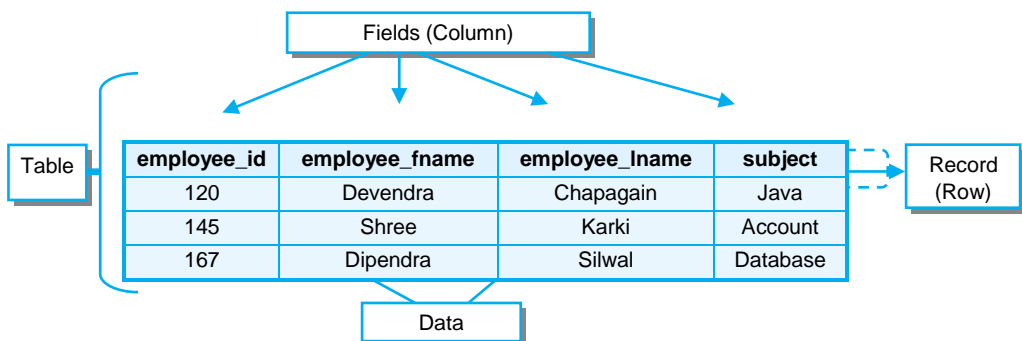
## Some terms related to Database

### Fields

A database field is a single piece of information from a record. A database record is a set of fields. Different types of database field include: Text, Date, Numeric, Yes/No, Dropdown, Image, Table, etc. In a database management system, a field can either be mandatory or optional. A mandatory field is a field in which you must enter data. If you don't enter data you will get an error message. An optional field is a field where you may or may not enter a data. You may leave the field blank and it does not get any error message. Each database has its own database field types.

### Records

A record is a collection of logically related fields or data items, with each field processing a fixed number of bytes and having a fixed data types. A record consists of values for each field. The grouping of data items can be achieved through different ways to form different records for different purposes. These records are retrieved or updated using programs.



## Files and File Systems

A file is a collection of related sequence of records. In many cases, all records in a file are of the same record type (each record having an identical format). If every record in the file has exactly the same size in bytes, the file is said to be made up of fixed-length records. If different records in the file have different sizes, the file is said to be made of variable-length records.

File-processing system is supported by a conventional operating system. The system stores permanent records in various files, and it needs different application program to extract records from the appropriate files and add record to appropriate files.

### Advantage of File Processing System

- Provides a useful historical perspective on. how to handle data.
- The characteristics of a file-based system helps in an overall understanding of design complexity of database system.
- Understanding the problems and knowledge of limitation inherent in the file based system helps avoid these same problems when designing database systems and thereby resulting in smooth transition.

### Disadvantages of File Processing System

- Excessive programming effort
- Data Inconsistency
- Limited data sharing
- Poor data control
- Inadequate data manipulation capabilities
- Data Redundancy (or duplication)
- Atomicity problems
- Security problems
- Integrity problems
- Program data dependence
- Data isolation
- Difficulty in accessing data
- Concurrent access anomalies

## Objects

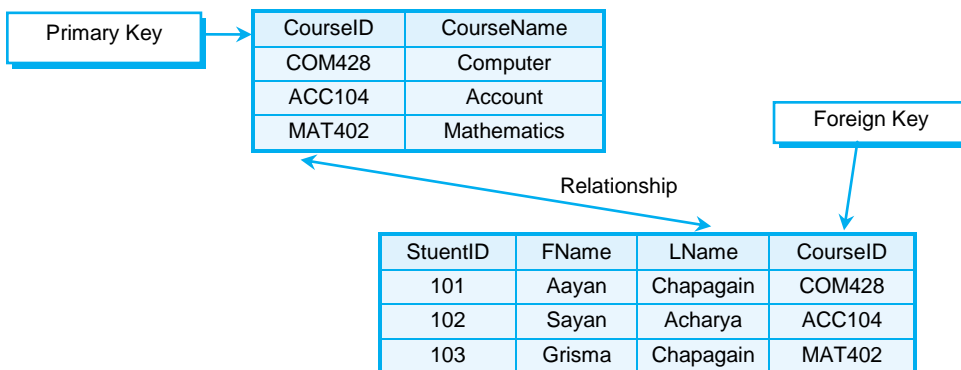
A database object in a relational database is a data structure used to either store or reference data. The most common object that people interact with is the table. Other objects are indexes, stored procedures, sequences, views etc. An object is an item that contains data, as well as the actions that read or process the data. A Student object, for example, might contain data about a student such as Student ID, First Name, Last Name, Address, Attendance Record, Marks and so on. Object oriented database (OODB) stores data in objects.

An object-oriented database stores unstructured data more efficiently than a relational database. Unstructured data includes photos, video clips, audio clips, and documents. When users query an object-oriented database, the results often are displayed more quickly than the same query of a relational database.

## Keys

Key is a field that uniquely identifies the records, tables or data. Key in a table allows us to establish the relation between multiple tables. Keys are also useful for finding the unique records or combination of records from a large database tables. In the relational data model there are many keys. Some of these keys are:

1. **Primary Key:** A primary key is one or more column(s) in a table used to uniquely identify each row in the table. Primary key cannot contain Null value. There can be only one primary key in a table.
2. **Foreign Key:** Foreign keys represent relationships between tables. A foreign key is a column whose values are derived from the primary key of some other table.



3. **Unique Key:** Unique key is one or more column(s) in a table used to uniquely identify each row in the table. The unique key cannot have duplicate values. It can contain one NULL value.
4. **Super Key:** A super key is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation.
5. **Candidate Key:** If a relational schema has more than one key, each is called a candidate key. All the keys which satisfy the condition of primary key can be candidate key. There can be any number of candidate keys that can be used in place of the primary key if required.
6. **Alternate Key:** Alternate keys are those candidate keys which are not the Primary key. There can be only one Primary key for a table. Therefore all the remaining Candidate keys are known as Alternate.

Alternate Key is also known as secondary Key that accepts null values and that can be submitted to the attribute in a tuple. Alternate key is used to identify columns in a table that can uniquely identify all the records of the table.

Emp_ID	Emp_Name	Gender	City	Email
101	Brayan	M	Pokhara	brayan@gmail.com
102	Samir	M	Chitwan	samir@gmail.com
103	Kishan	M	Kawasoti	kishan@gmail.com
104	Agraj	M	Gaindakot	agraj@gmail.com
105	Menuka	F	Butwal	menuka@gmail.com

Primary Key

Alternate Key

In the above employee table Emp\_ID is a primary key and other candidate key (Email) becomes the alternate key for the table

## Database Model

A database model is a relatively simple representation, usually graphical, of more complex real-world data structures. The function of database model is to understand the complexities of the real-world environment. Within the database environment, a database model represents data structures and their characteristics, relations, constraints, transformations, and other constructs with the purpose of supporting a specific problem domain.

The different types of database model are:

- Hierarchical Database Model
- Network Database Model
- Relational database model
- Entity Relationship Model
- Object Oriented Model

### 1. Hierarchical Model

In hierarchical model, its structure contains levels or segments. A segment is the equivalent of a file system's record type. Within the hierarchy, a higher layer is perceived as the parent of the segment directly beneath it, which is called the child. The hierarchical model depicts a set of one-to-many (1: M) relationships between a parent and its children segments.

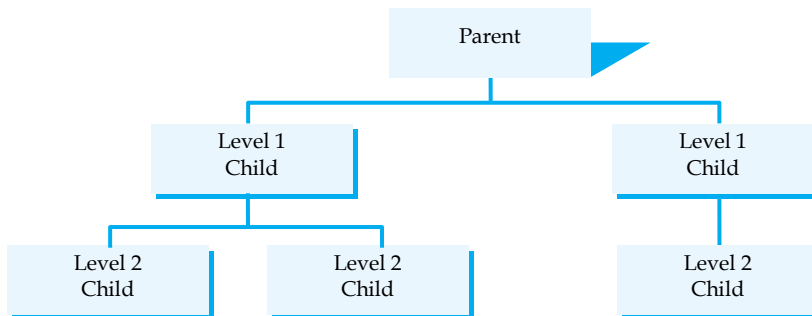


Figure: Hierarchical Model

### Advantages

- It promotes data sharing.
- Parent/Child relationship promotes conceptual simplicity.
- Database security is provided and enforced by DBMS.
- Parent/Child relationships promote data integrity.
- It is efficient with 1: M relationships.

### Disadvantages

- Complex implementation requires knowledge of physical data storage characteristics.
- Navigational system yields complex application development, management, and use; requires knowledge of hierarchical path.
- Changes in structure require changes in all application programs.
- There are implementation limitations (no multi parent or M:N relationships).
- There is no data definition or data manipulation language in the DBMS.
- There is a lack of standards.

## 2. Network Model

The network model was created to represent complex data relationships more effectively than the hierarchical model, to improve database performance, and to impose a database standard. In the network model, the user perceives the network database as a collection of records in 1: M relationships. However, unlike the hierarchical model, the network model allows a record to have more than one parent.

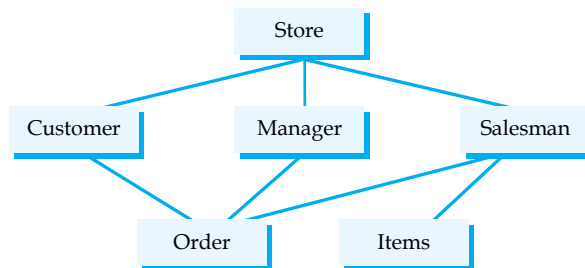


Figure: Network Model

- Conceptual simplicity is at least equal to that of the hierarchical model.
- It handles more relationship types, such as M: N and multi-parent.
- Data access is more flexible than in hierarchical and file system models.
- Data Owner/Member relationship promotes data integrity.
- There is conformance to standards - that meet the requirement of the standard.
- It includes data definition language (DDL) and data manipulation language (DML) in DBMS.

### Disadvantages

- System complexities limit efficiency – still a navigational system.
- Navigational system yields complex implementation, application development, and management.
- Structural changes require changes in all application programs.
- Structure is complicated than hierarchal model. Therefore it is difficult to handle and maintain.

### 3. Relational Model

The relational model represented a major breakthrough for both users and designers. To use an analogy, the relational model produced an “automatic transmission” database to replace the “standard transmission” databases that preceded it.

Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables. So, the tables are also called relations in the relational model.

Table name: Instructor

employee_id	employee_fname	employee_lname	subject
120	Devendra	Chapagain	Java
145	Shree	Karki	Account
167	Dipendra	Silwal	Database

Linked through employee\_id

Table name: Period\_details

room_no	period	subject	section	employee_id
BC 004	Second	Database	Four Sem	167
BC 207	Third	Account	First sem	145
BC 304	First	Java	Eight sem	120

In the above example, two tables are linked through employee\_id through instructor and period\_details. The tables are independent of one another; but can easily associate the data between tables. The relational model provides a minimum level of controlled redundancy to eliminate most of the redundancies commonly found in file systems.

The relational data model is implemented through a very sophisticated relational database management system (RDBMS). The RDBMS performs the same basic functions provided by the hierarchical and network DBMS systems, in addition to a host of other functions that make the relational data model easier to understand and implement.

### Advantages

- Structural independence is promoted by the use of independent tables. Changes in a table's structure do not affect data access or application programs.

- Tabular view substantially improves conceptual simplicity, thereby promoting easier database design, implementation, management, and use.
- Powerful RDBMS isolates the end user from physical-level details and improves implementation and management simplicity

### Disadvantages

- The RDBMS requires substantial hardware and system software overhead.
- Conceptual simplicity gives relatively untrained people the tools to use a good system poorly, and if unchecked, it may produce the same data anomalies found in file systems.
- It may promote “islands of information” problems as individuals and departments can easily develop their own applications.




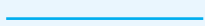
## 4. Entity-Relationship Model

Entity-Relationship models are normally represented in an entity relationship diagram (ERD), which uses graphical representations to model database components. This model is good to design a database, which can be later changed into tables in relational model.

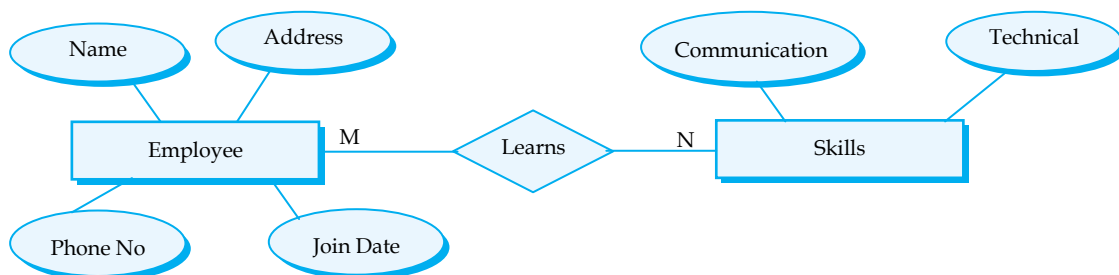
**Entity:** An entity is defined as anything about which data are to be collected and stored. An entity is represented in the ERD by a rectangle, also known as an entity box. Each entity is described by a set of attributes that describes particular characteristics of the entity.

**Relationships:** Relationships describe associations among data. Most relationships describe associations between two entities. When the basic data model components were introduced, three types of relationships among data were illustrated: one-to-many (1:M), many-to-many (M:N), and one-to-one (1:1). The ER model uses the term connectivity to label the relationship types.

Following notations are used in the ER Diagram:

Symbol	Representation	Description
	Rectangle	It represents the entity
	Ellipse	It represents the Attributes
	Diamond	It represents relationship
	Line	It represents the link between attribute and entity.

**Example:** an EMPLOYEE can learn many SKILLS and each SKILL can be learned by many EMPLOYEEs.



### Advantages

- Visual modeling yields exceptional conceptual simplicity.
- Visual representation makes it an effective communication tool.
- It is integrated with dominant relational model.

### Disadvantages

- There is limited constraint representation.
- There is limited relationship representation.
- There is no data manipulation language.
- Loss of information content occurs when attributes are removed from entities to avoid crowded displays. (This limitation has been addressed in subsequent graphical versions.)

## 5. Object-Oriented Model

In the object-oriented model, both data and their relationships are contained in a single structure known as an object. An Object-Oriented Model reflects a very different way to define and use entities. An object includes information about relationships between the facts within the object, as well as information about its relationships with other objects. An objects include data, various types of relationships, and operational procedures, the object becomes self-contained, thus making the object – at least potentially – a basic building block for autonomous structures.

### Advantages

- Semantic content is added.
- Visual representation includes semantic content.
- Inheritance promotes data integrity.

### Disadvantages

- Slow development of standards caused vendors to supply their own enhancements, thus eliminating a widely accepted standard.
- It is a complex navigational system.
- There is a steep learning curve.
- High system overhead slows transactions.

## Concept of Normalization

Normalization is a process for evaluating and correcting table structures to minimize data redundancies, thereby reducing the likelihood of data anomalies. The normalization process involves assigning attributes to tables based on the concept of determination.

Simply, Normalization is the concept of breaking down the larger table into small group of sub-tables so that data redundancy and dependency of data can be minimized. Normalization divides a big table into smaller tables and defines relationship between those small tables.

Normalization works through a series of stages called normal forms, they are:

- First Normal Form (1NF),
- Second Normal Form (2NF),
- Third Normal Form (3NF) and
- Fourth Normal Form (4NF)

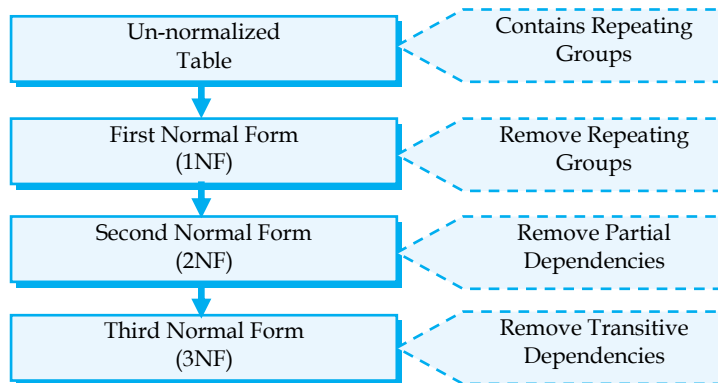


Figure: Steps in Normalization Process

### Advantages of Normalization

- Normalization Process helps to produce cost-effective database systems
- Normalization eliminate redundant (repetitive) data
- Normalization reduce the amount of storage needed to store the data
- Normalization ensures that the data duplication is minimized.
- Normalization avoid unnecessary data conflicts
- Normalization improves the database performance
- Normalization helps for faster searching of data

The following table is in de-normalized form:

Consider the schema and its tabular format with data:

Student\_Teacher (RN, Name, Address, Phone, DOB, Tid, Dept, Tname, Tphone)

RN	Name	Address	Phone	DOB	Tid	Dept	Tname	Tphone
1.	Devendra Chapagain	Nawalparasi	078540103	2050/10/17	110	Management	Dev	9867031614, 078540103
2.	Dipendra Silwal	Chitwan	056501260	2049/5/08	135	Science	Dip	9845345953, 056501260
3.	Tez Aryal	Chitwan	056620200	2051/11/26	140	Management	Tez	9855082985

This table is not a normalized table. The data field Tphone consist multiple data on it, this violation the rule of normalization.

- 1. First Normal Form (1NF):** The above table, the field Tphone holds multiple values in one cell. In this case the primary key is RN. With the design like this table, we can have the insert, update, delete and select anomalies. Note that only update and select anomalies deal with redundancy. They are problematic because the table contains redundant data. Insert and delete anomalies deal with incompleteness of the table.

A relation is said to be in First Normal Form (1NF) if and only if each attribute of the relation is atomic. Atomic means, the smallest piece of data which cannot be divided further. More simply, to be in 1NF, each column must contain only a single value and each row must contain the same columns. The purpose of the 1NF is to eliminate repeating groups of attributes in an entity. The relation to be in 1NF can be represented by the following schema and corresponding table.

After applying the rule of 1NF, we get the above denormalized table as follow:

Student\_Teacher (RN, Name, Address, Phone, DOB, Tid, Dept, Tname, Tphone)

RN	Name	Address	Phone	DOB	Tid	Dept	Tname	Tphone
1	DevendraChapagain	Nawalparasi	078540103	2050/10/17	110	Management	Dev	9867031614
1	DevendraChapagain	Nawalparasi	078540103	2050/10/17	110	Management	Dev	078520170
2	DipendraSilwal	Chitwan	056501260	2049/5/08	135	Science	Dip	9845345953
2	DipendraSilwal	Chitwan	056501260	2049/5/08	135	Science	Dip	056501260
3	Tez Aryal	Chitwan	056620200	2051/11/26	140	Management	Tez	9855082985

This table is in 1NF because every field of the table is atomic, i.e. small piece of data consists in the table field. Further, there are no duplication of rows or columns.

- 2. Second Normal Form (2NF):** In the above table, it is in 1NF, we still have the insert, update and delete anomalies. To remove the above anomalies, the table should be changed into 2NF. A relation is said to be in 2NF if the relation must first fulfill the requirements to be in First Normal Form and each non-keyattribute in the relation must be functionally dependent upon the primary key.

In the above example, the RN and Tid with Tphone field can be used as primary keys. Similarly, the corresponding key field depends on its primary keys like Name, Address, Phone and DOB depends on RN field and Dept, Tname and Tphone depends on Tid. To change the above table in to 2NF, we need to decompose the table into multiple tables. The relation to be in 2NF can be represented by the following schema and corresponding table:

Student(RN, Name, Address, Phone, DOB)

Teacher(Tid, Tname, Tphone)

Std\_Tea(RN, Tid)

RN	Name	Address	Phone	DOB
1	DevendraChapagain	Nawalparasi	078540103	2050/10/17
2	DipendraSilwal	Chitwan	056501260	2049/5/08
3	Tez Aryal	Chitwan	056620200	2051/11/26

Student Table

Tid	Tname	Tphone
110	Dev	9867031614
110	Dev	078540103
135	Dip	9845345953
135	Dip	056501260
140	Tez	9855082985

Teacher Table, where Tid and Tphone jointly work as primary key.

RN	Tid
1	110
2	135
3	140

Std\_Tea, Student and Teacher relational table

After decomposing to multiple tables, we must have some common values among these tables that enable us to join tables in queries. Otherwise, we would lose some information from table. For example, in last table Std\_Tea, both candidate keys are collected to maintain relationship between two tables. This process indicates the relational database.

- Third Normal Form (3NF):** The above tables are in 2NF, but still have Insert and Delete anomalies. To remove these anomalies, these tables should be changed into 3NF. A relation is said to be in 3NF, a table must be in 2NF and non-prime attribute of table is non-transitively dependent (i.e. directly dependent) on every candidate key. Additionally, all attributes that are not dependent upon the primary key must be eliminated.

For example, in above table Student (RN, Name, Address, Phone, DOB), Name, Address and Phone are fully depending on it primary key 'RN'. But the field Date Of Birth 'DOB' is depends on student name 'Name' not in primary key 'RN'. After applying the rules of 3NF, the Student tables will be decomposed into the following tables but other tables remain same because other supports the rule of 3NF. The schema and corresponding tables are as follows:

Student1(RN, Name, Address, Phone)

Student2(RN, DOB)

In this case, no relational table require because the primary key 'RN' contains in both tables and works as relationship between two tables.

RN	Name	Address	Phone
1	Devendra Chapagain	Nawalparasi	078540103
2	Dipendra Silwal	Chitwan	056501260
3	Tez Aryal	Chitwan	056620200

Student1 Table

RN	DOB
1	2050/10/17
2	2049/5/08
3	2051/11/26

Student2 Table

This schema is free from insert, update, delete and select anomalies. Still, there are some rules for decomposing the tables depending upon higher order normal forms. These normal forms are beyond the scope of this course design.

#### Another Example of Normalization Process:

##### Un-normalized Table:

- A table is un-normalized if it has multiple values in a cell.

Table: Employee

Emp_No	Name	Dept_No	Dept_Name	Skills
1	Bibek	101	IT	C, Java
2	Bishnu	101	IT	Linux, C#
3	Balkrishna	201	Humanities	Oracle, Java

**Description:** The above table is un-normalized table since it has multiple values in a single cell. This table should be changed to First Normal Form as:

##### First Normal Form

A table in 1NF; each attribute must be atomic.

Table: Employee

Emp_No	Name	Dept_No	Dept_Name	Skills
1	Bibek	101	IT	C
1	Bibek	101	IT	Java
2	Bishnu	101	IT	C#
2	Bishnu	101	IT	Linux
3	Balkrishna	201	Humanities	Oracle
3	Balkrishna	201	Humanities	Java

**Description:** The above table is in First Normal Form. There are no multiple values in a single cell. In the above table, Name, Dept\_No and Dept\_Name are functionally dependent on Emp\_No. So it can be converted into Second Normal Form as:

### Second Normal Form

- Each attribute must be functionally dependent on Primary key.

**Table: Employee**

Emp_No	Name	Dept_No	Dept_Name
1	Bibek	101	IT
2	Bishnu	101	IT
3	Balkrishna	201	Humanities

**Table: Skills**

Emp_No	Skills
1	C
1	Java
2	C#
3	Linux
3	Oracle

Description: The above table is in Second Normal Form. In table Employee Dept\_No and Dept\_Name are functionally dependent on emp\_No. However, department can be considered as separate key. So It can be further decomposed into Third Normal Form.

### Third Normal Form

**Table: Employee**

Emp_No	Name	Dept_No
1	Bibek	101
2	Bishnu	101
3	Balkrishna	201

**Table: Skills**

Emp_No	Skills
1	C
1	Java
2	C#
3	Linux
3	Oracle

**Table: Department**

Dept_No	Dept_Name
101	IT
102	Humanities

**Descriptions:** The above tables are now in Third Normal Form.

## Another Example of Normalization Process

Table: Course\_Info (Un-normalized Table)

Faculty	Instructor	Course Preferences	
		Course	Course_Dept
Computer Engineering	Devendra Chapagain	CMP 332	Computer Science
		ELE 221	Electrical and Electronic
		CMP 388	Computer Science
	Dipendra Silwal	CMP 103	Computer Science
		CMP 104	Computer Science
		CMP 383	Computer Science
	Tez Aryal	ELX 226	Electrical and Electronics
		CMP 331	Computer Science
		CMP 224	Computer Science

### In 1NF:

Instructor	Course	Faculty	Course_Dept
Devendra Chapagain	CMP 332	Computer Engineering	Computer Science
Devendra Chapagain	ELE 221	Computer Engineering	Electrical and Electronic
Devendra Chapagain	CMP 388	Computer Engineering	Computer Science
Dipendra Silwal	CMP 103	Computer Engineering	Computer Science
Dipendra Silwal	CMP 104	Computer Engineering	Computer Science
Dipendra Silwal	CMP 383	Computer Engineering	Computer Science
Tez Aryal	ELX 226	Computer Engineering	Electrical and Electronic
Tez Aryal	CMP 331	Computer Engineering	Computer Science
Tez Aryal	CMP 224	Computer Engineering	Computer Science

### Second Normal Form (2NF)

Course	Instructor	Room	Time	Enroll_Unit
CMP 332	Devendra Chapagain	BC 304	0600 Hrs	80
ELE 221	Devendra Chapagain	BE 204	0715Hrs	60
CMP 103	Dipendra Silwal	BC 205	0800HRs	75
CMP 104	Dipendra Silwal	BC 208	0900Hrs	90
ELX 226	Tez Aryal	BE 207	0600Hrs	140
CMP 224	Tez Aryal	BC 302	0100Hrs	65

### Instructor Relation in 2NF:

Course	Instructor	Enroll_Unit
CMP 332	Devendra Chapagain	80
ELE 221	Devendra Chapagain	60
CMP 103	Dipendra Silwal	75
CMP 104	Dipendra Silwal	90
ELX 226	Tez Aryal	140
CMP 224	Tez Aryal	65

a.

Course	Room
CMP 332	BC 304
ELE 221	BE 204
CMP 103	BC 205
CMP 104	BC 208
ELX 226	BE 207
CMP 224	BC 302

b.

Room	Time
BC 304	0600 Hrs
BE 204	0715Hrs
BC 205	0800HRs
BC 208	0900Hrs
BE 207	0600Hrs
BC 302	0100Hrs

c.

Third Normal Form (3NF)

Example:

Course	Instructor	Room	Time	Enroll_Unit
CMP 332	Devendra Chapagain	BC 304	0600 Hrs	80
ELE 221	Devendra Chapagain	BE 204	0715Hrs	60
CMP 103	Dipendra Silwal	BC 205	0800HRs	75
CMP 104	Dipendra Silwal	BC 208	0900Hrs	90
ELX 226	Tez Aryal	BE 207	0600Hrs	140
CMP 224	Tez Aryal	BC 302	0100Hrs	65

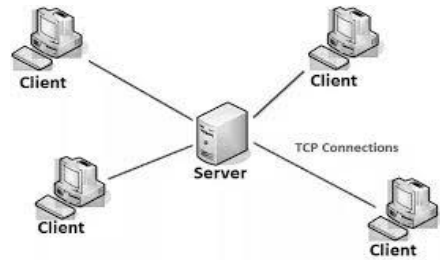
In 3NF:

Course	Enroll_Unit
CMP 332	80
ELE 221	60
CMP 103	75
CMP 104	90
ELX 226	140
CMP 224	65

## Compare between Centralized and Distributed Database

### Centralized Database

As shown in the above figure, centralized database works in a client-server system. The centralized database has one central computer, called database server, to store all the data and files and it provides services to all the clients in the networks. Only the central computer or database server is responsible for processing the data. Data retrieve is not very difficult and security also is not so crucial part as a DBA is there as a controller of the whole database. All the rules and security mechanism have been applied only to the database server. All the clients are responsible for just presenting the required records to the users in the format they desire. The clients never process the data. It is suitable for small organization which has different departments.



### Advantages of Centralized Database

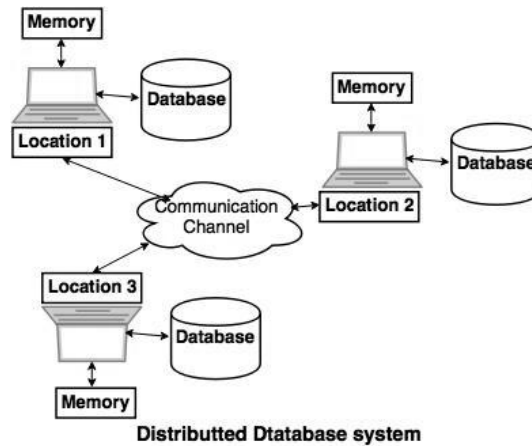
- Low cost to set up.
- High performance.
- Centralization of all data in a single computer called server.
- Easier to manage to manage and manipulate data and database.
- High security as a single Database Administrator can control the whole database system.
- Suitable for small organization with different departments.
- It is easier to manage and manipulate the data as data is stored only in a server computer.
- Easier data access.

### Disadvantages of Centralized Database

- Cannot cover larger area and not suitable for large organization.
- Database is location dependent, cannot be accessed from other places.
- It does not support globalized connection.

### Distributed Database

Distributed database system is a set of database stored on multiple computers that appears to applications as a single database. The user can simultaneously access and modify the data in several databases in a network. The computers in a distributed system communicate with each other through various communication media, such as high speed buses or telephone line. These computers neither share the main memory nor a clock cycle of processor, although to work properly many applications on different computers might have to synchronize their clicks.



### Distributed Database System

The main difference between centralized database and distributed database is that, in the centralized database, the data resides in one single centralized computer, while in the distributed database. The data is stored in several sets under the control of local distributed DBMS components which are under the control of distributed database system.

As shown in above figure, distributed database system is located in many different geographical locations and they are connected with each other through the internet. The distributed database system is the interconnected form of heterogeneous centralized database systems. It is useful for very large organizations, which are spread all around the world for similar purpose with similar data sets and operations. Data communication is more difficult in it because databases with heterogeneous platforms have been linked through the internet. The data security is very much crucial part because it can be hacked or damaged during the data transmission. Since there are large numbers of users, it is very difficult to set appropriate permission to them. Location independence is its main feature so that the user can access the required data from any location.

### Advantages of Distributed Database

- Data sharing and distributed control all over the world.
- Improved reliability for users.
- Improved availability of data.
- Economy on operation and data sharing.
- Modular growth can support.

### Disadvantages of Distributed Database

- Higher software development cost.
- Greater potential for bugs and hacked.
- Increased processing overhead for client and server computers.

- More Complexity in database design.
- Less security model because data may travel continent to continent.
- More difficult for general integrity

### Difference between Centralized Database and Distributed Database

Basis of Comparison	Centralized Database	Distributed Database
Location	It is managed by a single machine or system which is at a single location	It is spread and split up across various storage device locations
Maintenance	Easy to maintain	Difficult to maintain
Efficiency	Less Efficient	More Efficient
Failure	Entire data gets lost	Still able to access other databases
Response Speed	Slow	Fast
Communication Cost	High	Low

### Difference between DBMS and RDBMS

S.N.	DBMS	RDBMS
1	In DBMS relationship between two tables or files are maintained programmatically.	In RDBMS relationship between two tables or files can be specified at the time of table creation.
2	DBMS does not support client/server architecture.	Most of the RDBMS supports client/server architecture.
3	DBMS does not support distributed database.	Most of the RDBMS supports distributed database.
4	Each table is given an extension in DBMS.	Many tables are grouped in one database in RDBMS.
5	DBMS allows only one person to access the database at any given time.	RDBMS allows multiple users simultaneous access to the database.
6	Example: MS-Access, FoxPro, etc.	Example: Oracle, SQL Server, etc.

## Database Security

Database is an organized collection of related data to provide the right information at right time. Collected data are the basic element of the database to provide meaningful and useful information for the users and the management. Data plays very important roles to give right information at right time, to take right decision and even to make right plans for future. The data stored in the database should always remain safe, correct and authentic. There should not be any damage or harm to the data in the database.

Information security is the protection of information against unauthorized disclosure, alteration or destruction. Database security is the protection of information that is maintained in a database. It deals with ensuring only the 'right people' get the right to access the 'right data'. By right people, we mean those people who have the right to access or update the data that they

are requesting to access or update with the database. This should also ensure the confidentiality of the data. For example, the personal information of the employees should be accessible only to the authorities concerned and not to everyone.

Thus, one of the concepts of database security is primarily a specification of access rules about who has what type of access to what information. In a wide concept, data security also covers physical security such as data file protection from heat, dust, power failure, theft etc. The important objective of data security is to prevent data loss, misuse of data and disclosure of the secret data.

The methods of data security are as follows:

- By doing the regular backup of database in disks, tapes, optical disks etc. in order to prevent from the accidental loss.
- By using password log in system to prevent data from unauthorized access to the database.
- By specifying the specific roles to every user of the database for granting the appropriate permission to them.
- Making physical prevention by using stabilizer and UPS to supply a regular power through which we can prevent hardware and software from high electricity voltage and irregular power supply.
- By keeping the system in safe room or place with lock and key or under the supervision of watchman to prevent from theft.
- By implementing software protection like antivirus, firewalls, antispyware etc.

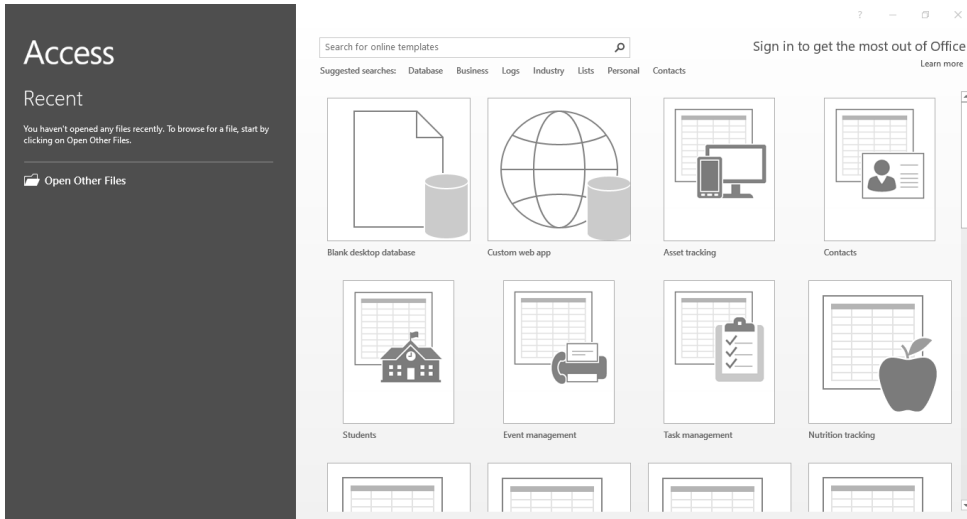
## Design Database using Microsoft Access

Microsoft Access is one of the most popular DBMS Software which is generally available in all PCs in which Microsoft Office package has installed. It is a single user DBMS. It is suitable for developing the software in desktop PC or laptop and it can be run by a single user at a time.

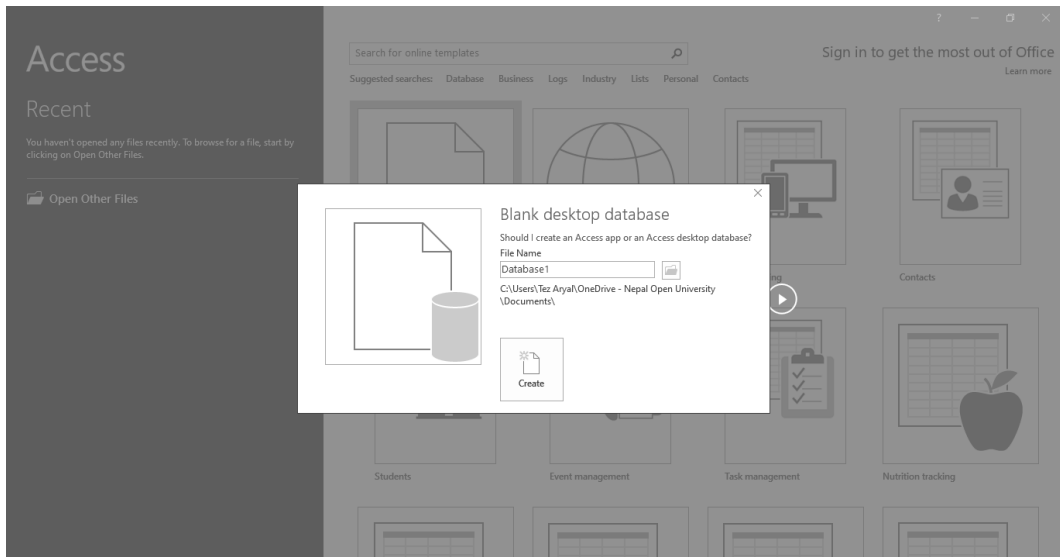
### Create a Database

Creating a database is the initial process of every database in computer. To create a database in Microsoft Access, you can follow the following steps:

- i. Start Microsoft Access from Program menu. (You can type 'msaccess' in Search box to start MS Access). Then you will see the following windows.



- ii. Click on Blank Desktop Database then specify the name of database and its location to save.
- iii. Click on Create.



## Create Tables and Fields and Its Properties

Table is a basic object of a database to hold the data element. So, you should design the table very efficiently so that it can hold all the required data.

A table contains data about a particular subject, such as employees or products. Each *record* in a table contains information about one item, such as a particular employee. A record is made up of *fields*, such as name, address and telephone number. A record is also commonly called a row, and a field is also commonly called a column.

Field or Column

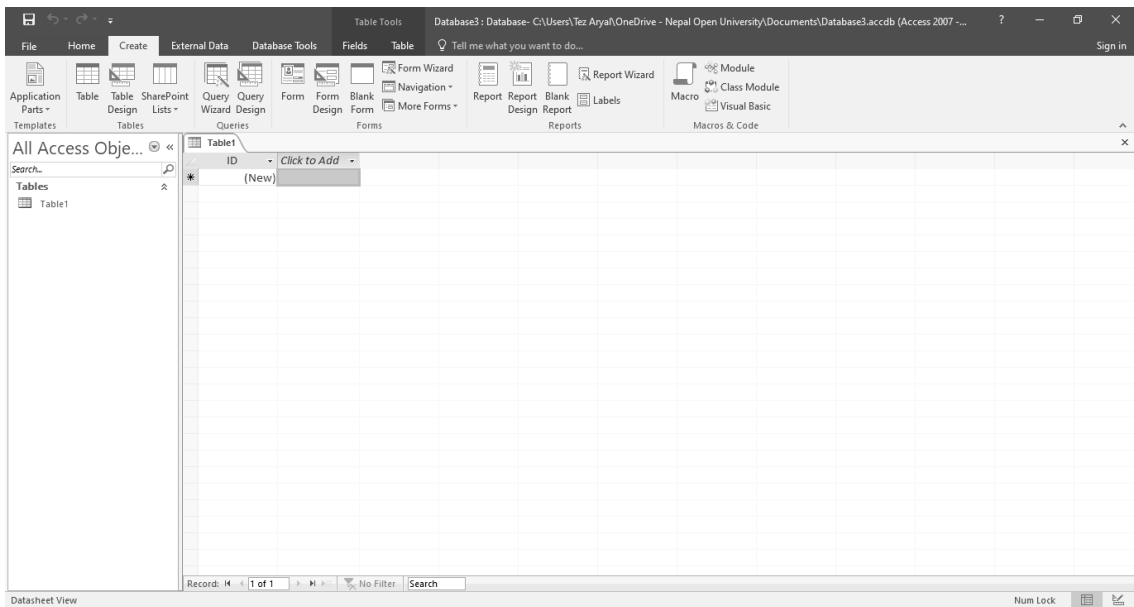
ID	E_Name	E_Departma
1	Ram Bdr Thapa	Admin
2	Dipendra Silwal	IT
3	Dev Chapagain	Plus 2

Record Row

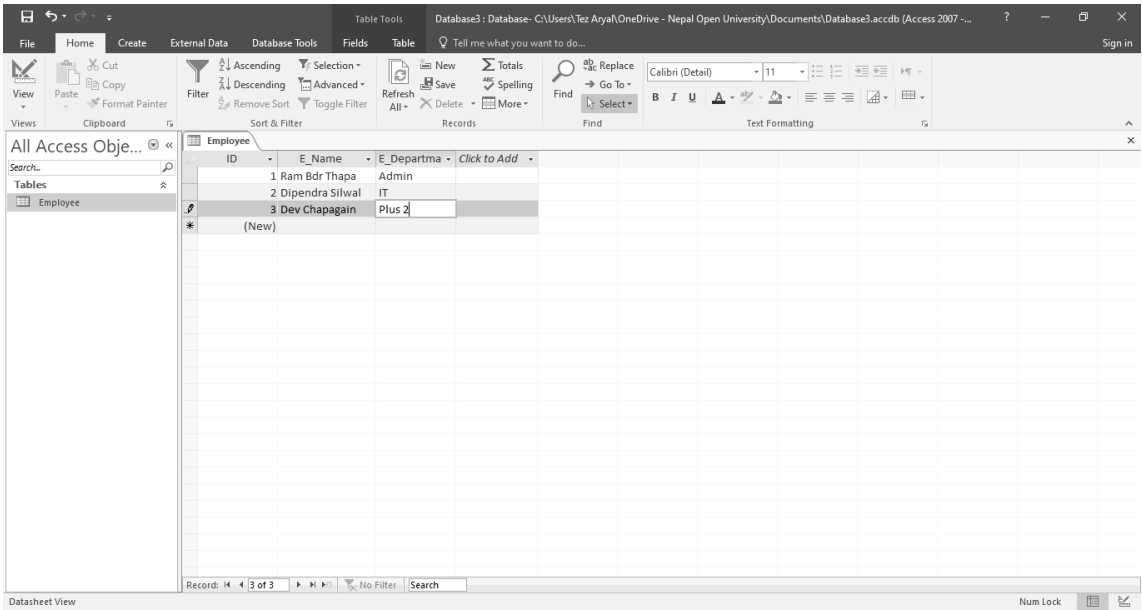
In Microsoft Access, you can create table generally by using a 'Datasheet View' or 'Design View'.

### Steps to Create Table in Datasheet View

- i. Open the database in which you want to create table.
- ii. Click on 'Create' tab → Click on 'Table' icon.



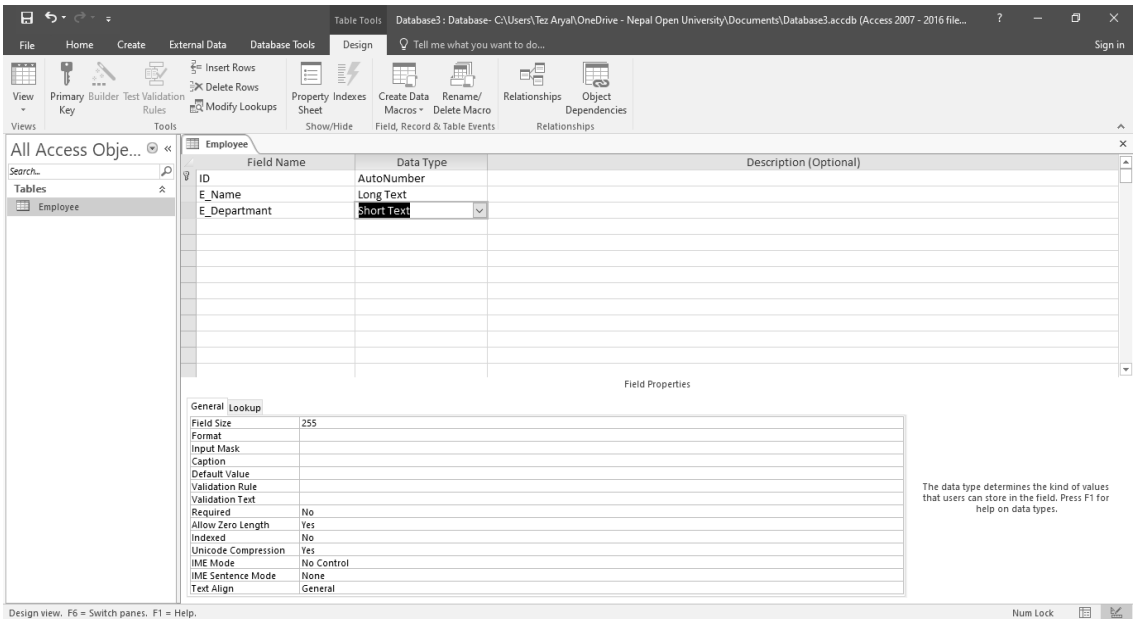
- iii. Select 'Datasheet' → Type the data which you want to store and if you want, you can change the name of the field heading by double clicking on it.



iv. Save the table and Name the table.

### Steps to Create Table in Design View

- i. Open the database in which you want to create table.
- ii. Click on 'Create' tab → Click on 'Table' icon.
- iii. Select Design View. You will see the following window.



- iv. Then type the name of the field, select appropriate data type.
- v. Save the table and change it into Datasheet view and start typing the data in the table.

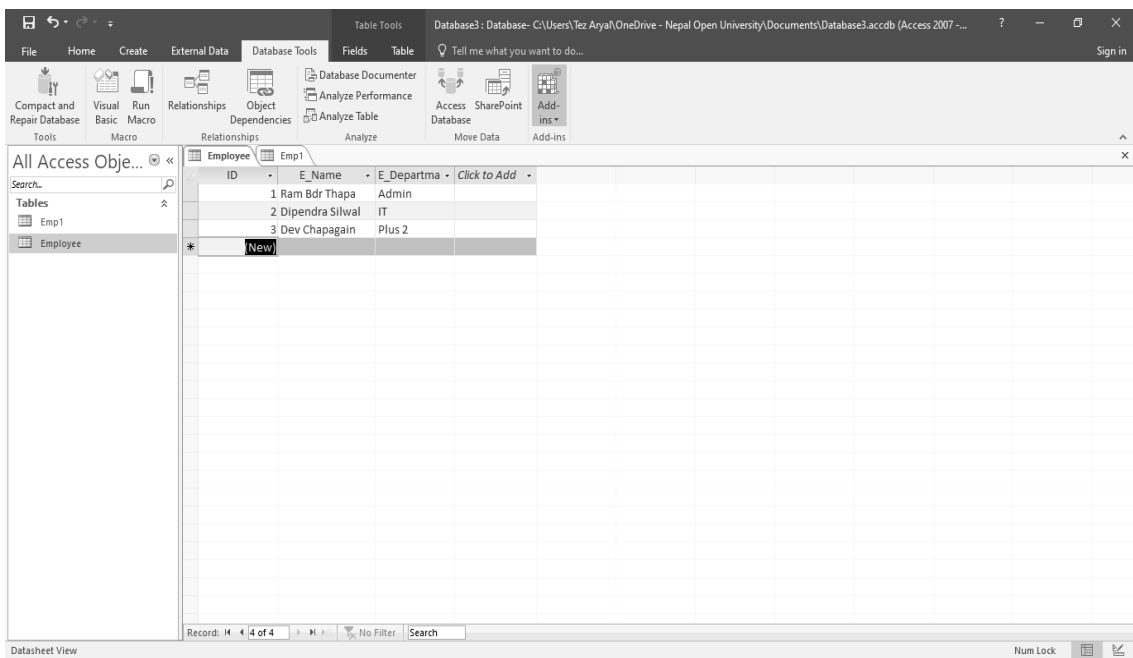
**[Note:** You can set the 'Primary Key' in appropriate field by clicking on the Primary key icon to make the values of that field unique. You can use only one Primary key per table.]

## Create a Relational Databases

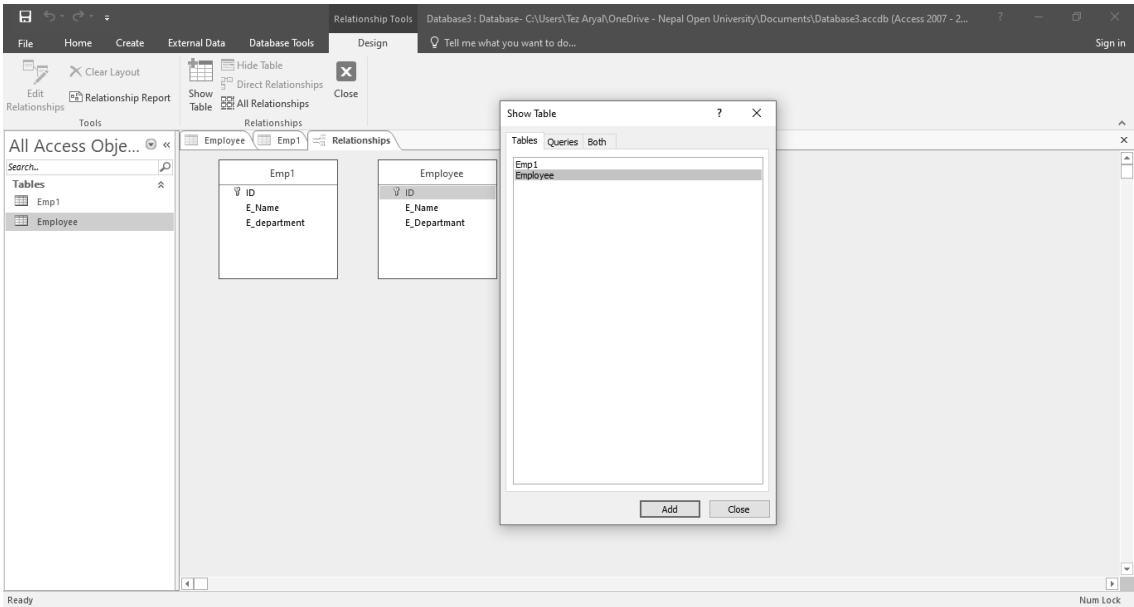
Relationship is a link between two tables in a database. If set properly, it also enables to maintain referential data integrity between the tables in database. To create a relationship between, tables you can go through the following steps:

**[Note:** Both tables should have a common field with common data type to create relationship between them.]

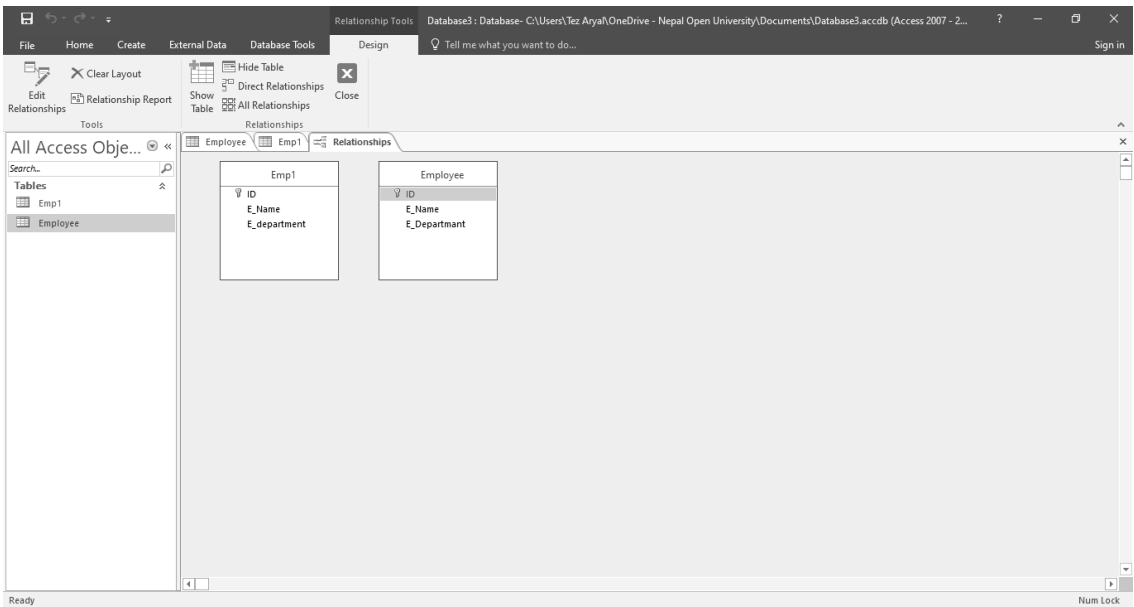
- i. Open the database in which you want to create relationship between tables.
- ii. Select the Database Tools tab and click on 'Relationship' icon.



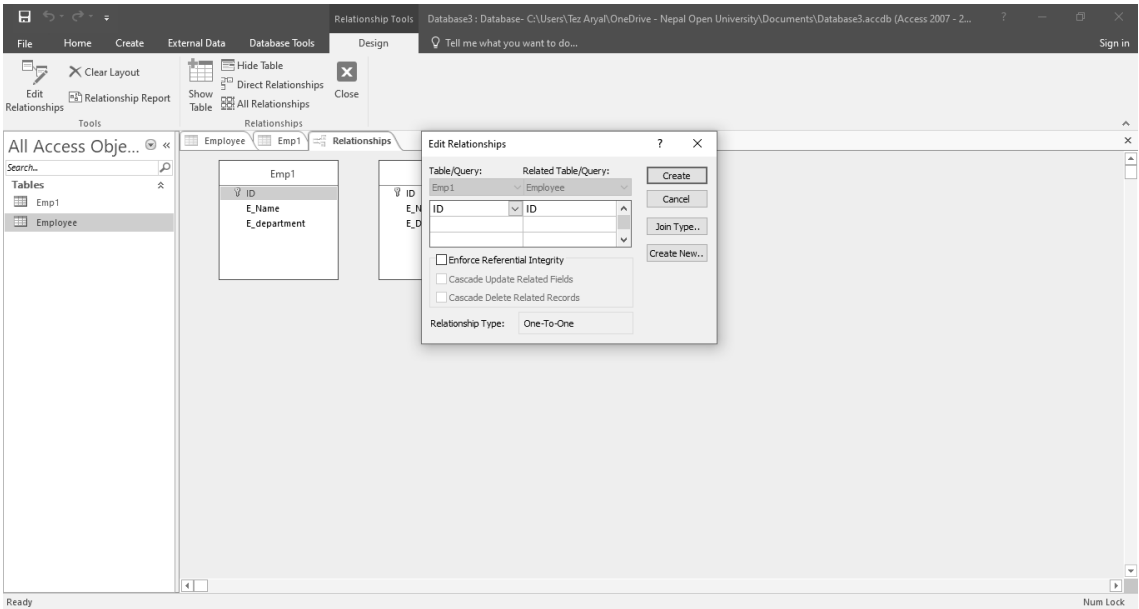
- iii. Click on 'Relationship' icon.



- iv. Select the required table name and click on Add and after adding all the required tables, click on Close.

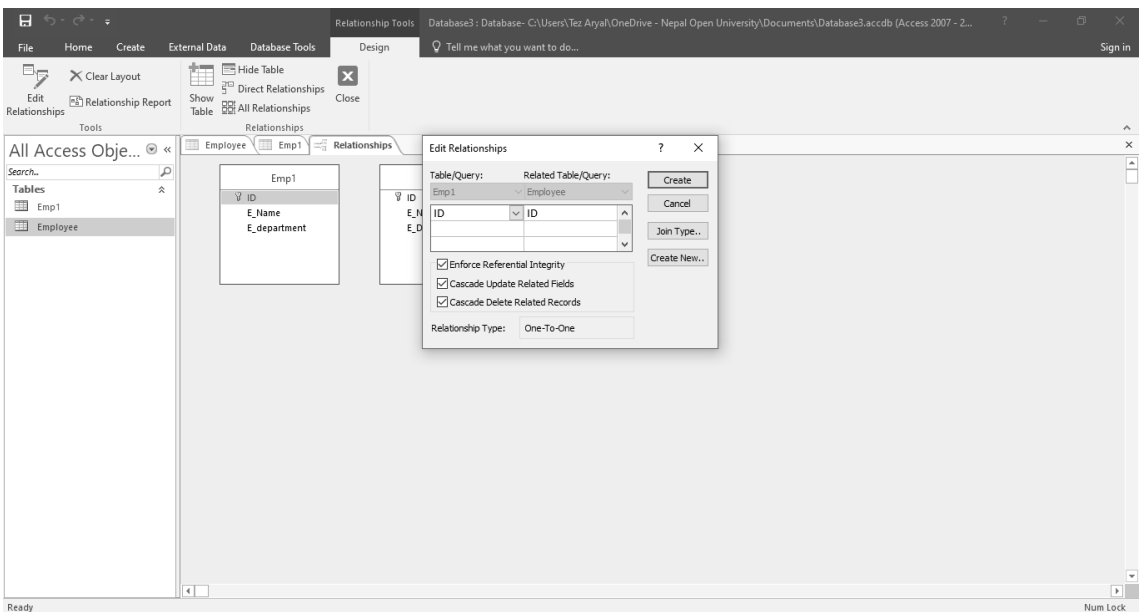


- v. Then drag the primary key field from master table and drop it to the common field in the child table.



vi. Tick the following:

- Enforce Referential Integrity.
- Cascade Update Related Fields.
- Cascade Deleted Related Fields.



vii. Click on Create.

## Create and Run Queries

Query is another important database object in Microsoft Access. It is used to display the specific records from the database. Query also enables you to append, delete or modify the specific records from their base tables in the database. In Microsoft Access, you can find the following types of queries.

1. **Select Query:** A select query is a type of database object that shows information in Datasheet view (Datasheet view: A window that displays data from a table, form, query, view, or stored procedure in a row-and-column format. In Datasheet view, you can edit fields, add and delete data, and search for data.). A query can get its data from one or more tables, from existing queries, or from a combination of the two. The tables or queries from which a query gets its data are referred to as its record source.
2. **Append Query:** An append query adds a set of records (rows) from one or more source tables (or queries) to one or more destination tables. You can also use append queries to:
  - Append fields that are based on criteria. For example, you might want to append only the names and addresses of customers who have outstanding orders.
  - Append records when some of the fields in one table don't exist in the other table. For example, suppose that your Customers table has 11 fields, and the fields in the Clients table in another database match 9 of your 11 fields. You can use an append query to add only the data in the matching fields and ignore the others.
3. **Make Table Query:** A make table query retrieves data from one or more tables, and then loads the result set into a new table.
4. **Update Query:** You use update queries to add, change, or delete the data in one or more existing records. You enter a select criterion and an update criterion. Update queries can accept multiple criteria, allow you to update a large number of records in one pass, and allow you to change records in more than one table at one time.
5. **Delete Query:** Delete queries remove all of the data in each field, along with the key value that makes a record unique. When you run the query, it always removes the entire row from the table.
6. **Cross Tab Query:** The query which aggregates data by two sets of values – one displayed down left side of the datasheet and another across the top.

### How to Create a Select Query?

You can create queries by using 'Query wizard' or 'Query Design View'.

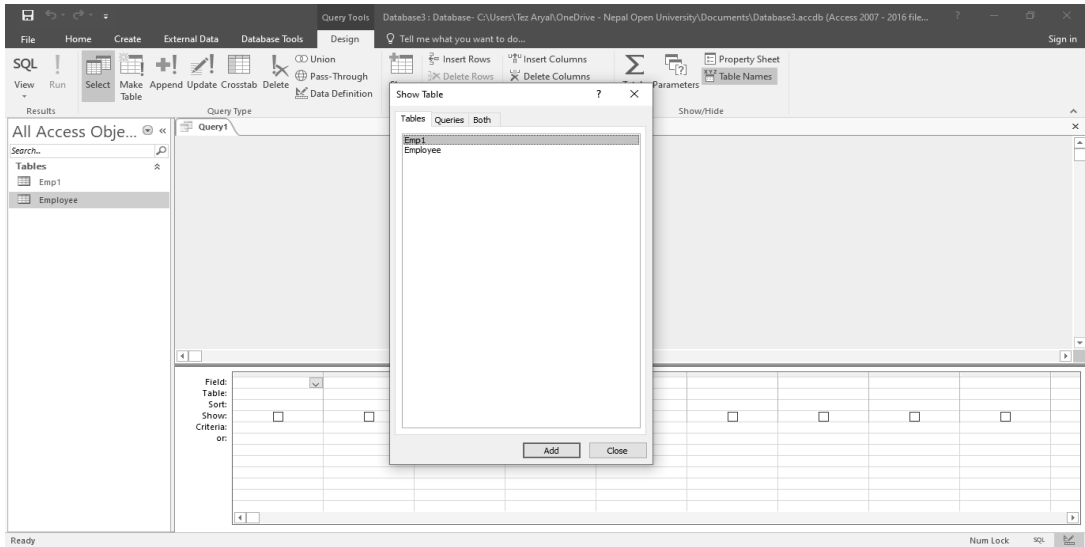
#### By using Query Wizard

This is the simple and easy method of creating 'Select Query' or 'Crosstab Query'. You can click on Query Wizard and the wizard guides you what to select and do accordingly as your requirements. After finishing the required steps, the wizard finally presents you the required result of the queries.

### By Using Query Design View

To create query by using this method, you need to follow the following steps:

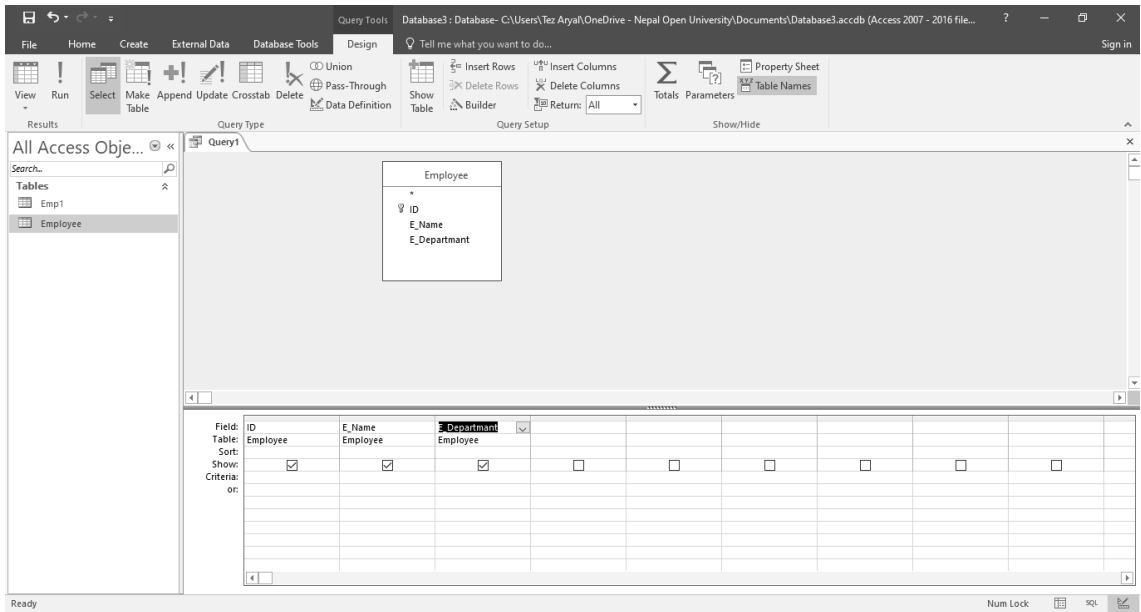
- i. Click on Create tab in Microsoft Access.
- ii. Then click on Query Design tab as shown in the figure.



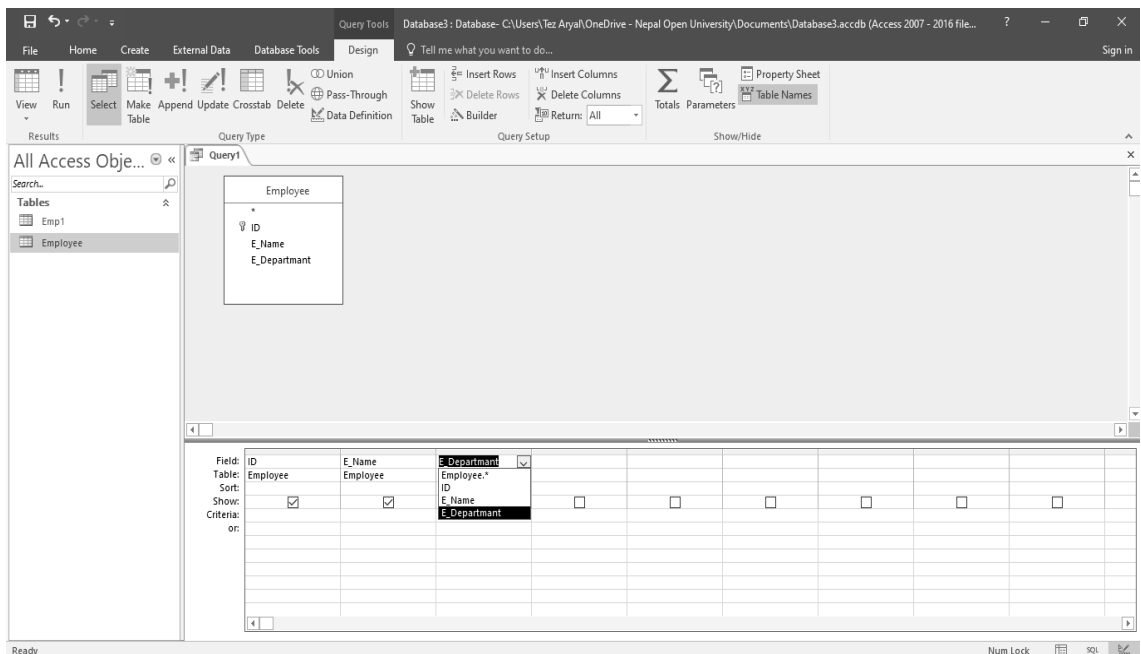
- iii. Select the required table and click on Add button and after finishing adding all the required tables, click on Close button.




- iv. Select the name of the fields in the query pane which you want to display in the query like below.



- iv. If necessary, then specify the necessary criteria below the required field in query pane. For example, if you want to display the records of those employees who are from 'IT Department' district only, then you can type 'IT' below 'E\_Deptant' filed as below.



- v. Then, finally you can save the query, name the query and run it by clicking on the 'Run' icon  from the design tab.

## Working with Forms

Form is another important database object in Microsoft Access. It allows you to enter, edit or display record from a table or a query. You can also use forms to control access to data i.e. which fields or rows of the table or query is to display. For example, if the English teacher wants to see only the marks of his/her subject from the resultsheet rather than marks of all subject, then you can make forms to view them the specific records they desire from the base tables. You can also add buttons and other functionality to a form to automate frequently performed actions.

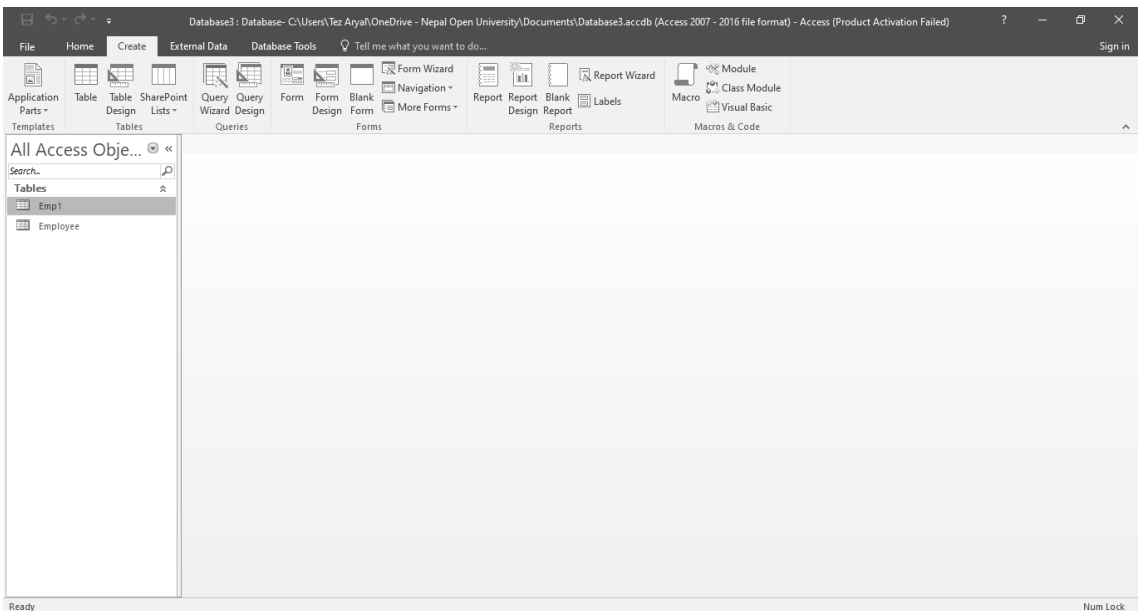
It is the graphical representation of table so that it is easy to manage the data in the underlying basic table. You can think of forms as windows through which people see and reach your database. An effective form speeds the use of your database, because people do not have to search for what they need. A visually attractive form makes working with the database more pleasant and more efficient, and it can also help prevent incorrect data from being entered.

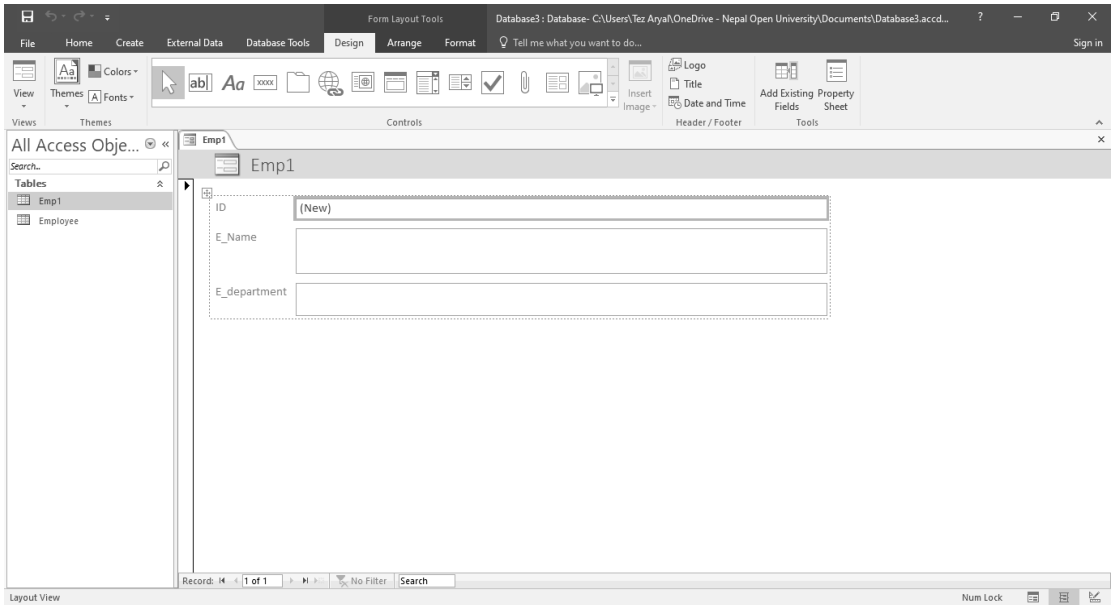
Microsoft Office Access 2007 gives you new tools to help you create forms quickly, and provides new form types and features that improve the usability of your database.

## How to Create Forms?

You can use the following steps to create form.

- i. Click on 'Create' tab in the database.
- ii. Select the table name or query name and click on 'Form' icon.





## Generate Reports

Report is another important object of database in Microsoft Access 2007. You can use report to display the required records from the table or query and make ready especially for printing.

A report consists of information that is pulled from tables or queries, as well as information that is stored with the report design, such as labels, headings, and graphics. The tables or queries that provide the underlying data are also known as the report's record source. If the fields that you want to include all exist in a single table, use that table as the record source. If the fields are contained in more than one table, you need to use one or more queries as the record source. Those queries may already exist in your database, or you may need to create new queries specifically to fit the needs of your report.

## Create a Report by Using the Report Wizard

You can use the Report Wizard to be more selective about what fields appear on your report. You can also specify how the data is grouped and sorted, and you can use fields from more than one table or query, provided you have specified the relationships between the tables and queries beforehand.

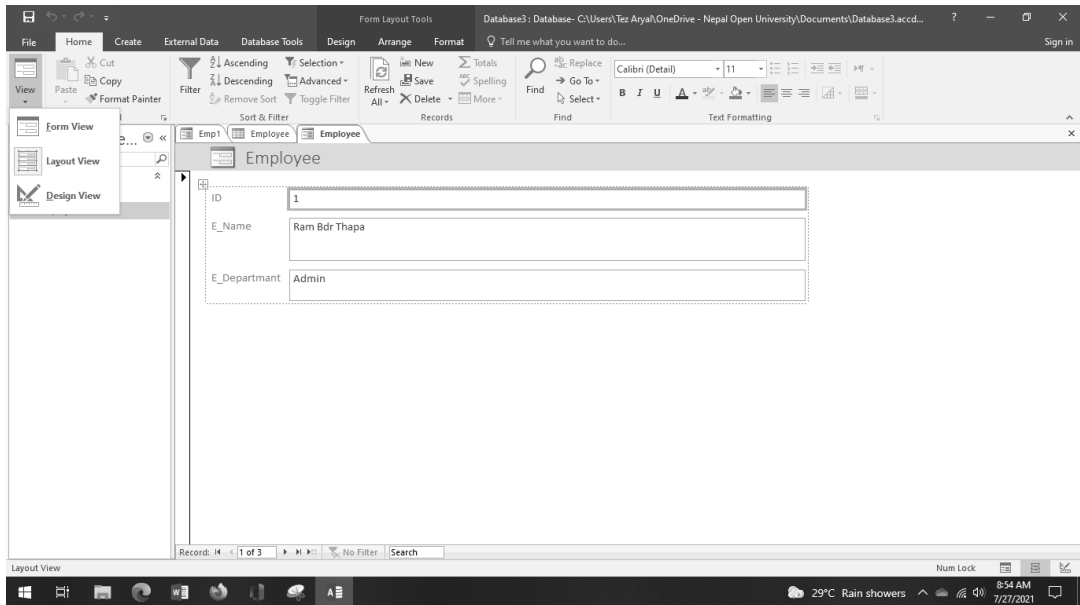
- On the **Create** tab, in the **Reports** group, click **Report Wizard**.
- Follow the directions on the Report Wizard pages. On the last page, click **Finish**.
- When you preview the report, you see the report as it will appear in print.

## Formatting Forms and Reports

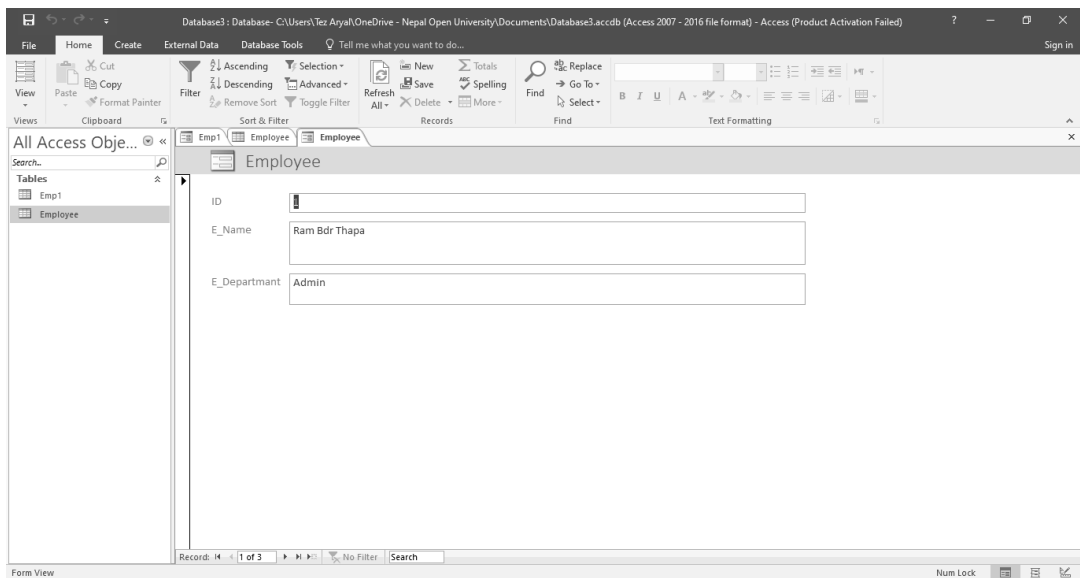
You already have the idea of how to create forms and reports in Microsoft Access database. Forms and reports should be very attractive so that users do not feel bored while working in the database. Thus, formatting forms and report enables you to create very attractive form and report.

## Formatting Forms

To apply the different formatting options in **form**, you need to open the form in Form Design view from the Home tab.

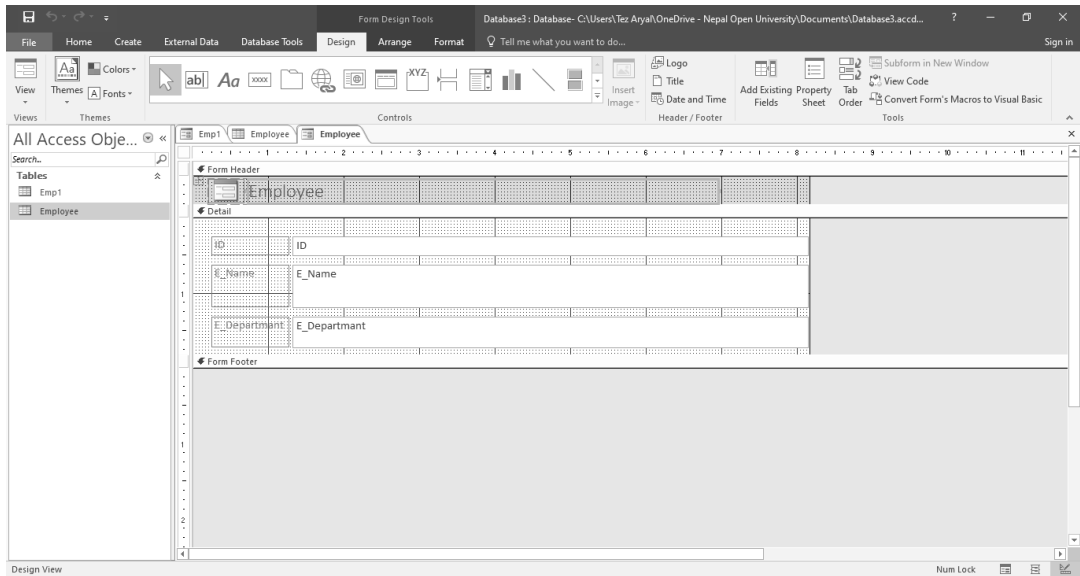


Then you will see the form as follows:

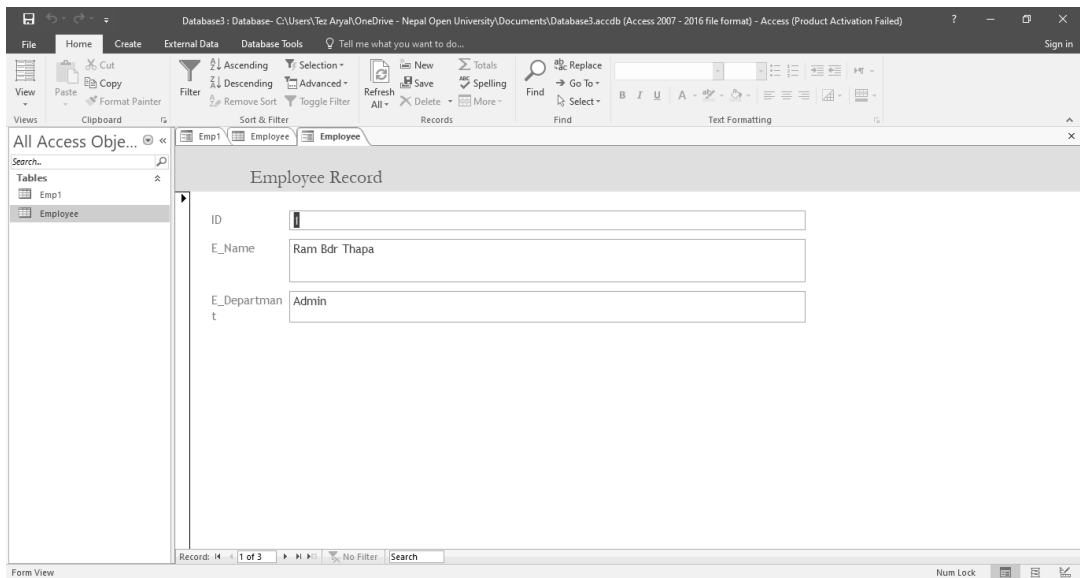


At the top of the window, you will see one more tab called 'Design' with many formatting commands on its ribbon.

Then you can apply the formatting options ranging from Fonts, Gridline, Controls and Properties. After applying some formatting effects, the above form looks like as follows:

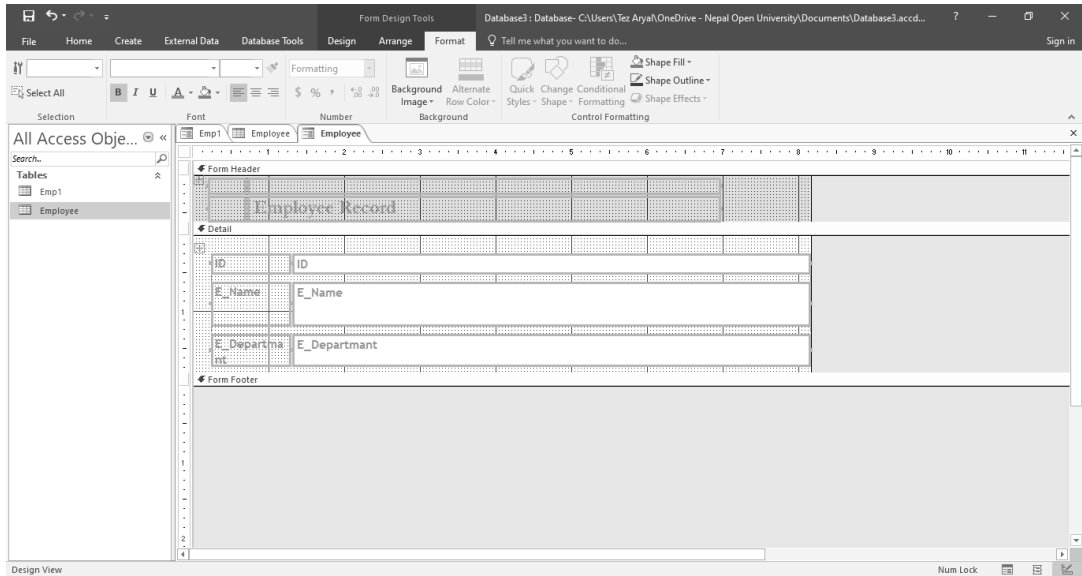


When you view the same form in Form View, it looks as follows:



## Formatting Report

Similarly, you can apply many formatting effects to the report as well. To apply the formatting effects in report in Microsoft Access, you need to open the report in 'Design View' from the 'Home' tab like as follows:



Then you will see the 'Design' tab with many formatting options like 'Font', 'Gridline', 'Control', 'Property' etc. and apply the appropriate options as required.

### Points to Remember

- Data are the raw or isolated facts and Information is the result of processing raw data to reveal its meaning.
- A database is a shared, integrated computer structure that stores a collection of end user data, i.e. raw facts of interest to end user and metadata (data about data), through which the end user data are integrated and managed.
- A database management system (DBMS) is a collection of programs that manages the database structure and controls access to the data stored in the database.
- A database system provides a data-definition language to specify the database schema and a data manipulation language to express database queries and updates.
- Data Definition Language (DDL) that allows a database administrator to define the database structure, schema, and subschema.
- Data Manipulation Language (DML) that allows an end user to manipulate the data in the database.
- There are five different types of database model: Hierarchical Model, Network Model, Relational Model, Entity-Relationship Model and Object-Oriented Model.
- Normalization is a process for evaluating and correcting table structures to minimize data redundancies, thereby reducing the likelihood of data anomalies. There are different types of Normalization. They are: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF) and Fourth Normal Form (4NF).

- A relation is said to be in First Normal Form (1NF) if and only if each attribute of the relation is atomic.
- A relation is said to be in 2NF if the relation must first fulfill the requirements to be in First Normal Form and each non-key attribute in the relation must be functionally dependent upon the primary key.
- A relation is said to be in 3NF, a table must be in 2NF and non-prime attribute of table is non-transitively dependent (i.e. directly dependent) on every candidate key. Additionally, all attributes that are not dependent upon the primary key must be eliminated.
- A record is a collection of logically related fields or data items, with each field processing a fixed number of bytes and having a fixed data types.
- A file is a collection of related sequence of records.
- The system stores permanent records in various files, and it needs different application program to extract records from the appropriate files and add record to appropriate files.
- Key is a field that uniquely identifies the records, tables or data. In the relational data model there are many keys.
- A primary key is one or more column(s) in a table used to uniquely identify each row in the table.
- A foreign key is a column whose values are derived from the primary key of some other table.
- Unique key is one or more column(s) in a table used to uniquely identify each row in the table.
- A super key is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation.
- If a relational schema has more than one key, each is called a candidate key. All the keys which satisfy the condition of primary key can be candidate key.
- The centralized database has one central computer, called database server, to store all the data and files and it provides services to all the clients in the networks.
- Distributed database system is a set of database stored on multiple computers that appears to applications as a single database.
- The computers in a distributed system communicate with each other through various communication media, such as high speed buses or telephone line.
- Information security is the protection of information against unauthorized disclosure, alteration or destruction. Database security is the protection of information that is maintained in a database. It deals with ensuring only the 'right people' get the right to access the 'right data'.
- Structured Query Language (SQL) is a language that provides an interface to relational database systems.

## ✓ Multiple Choice Questions

1. This key that uniquely identifies each record is called:
  - a. Primary Key
  - b. Key Record
  - c. Unique Key
  - d. Field Name
2. A data dictionary is a repository that manages \_\_\_\_\_
  - a. Memory
  - b. Metadata
  - c. Spell Checker
  - d. Data Validator
3. Which of the following is generally used for performing tasks like creating the structure of the relations, deleting relation?
  - a. DML(Data Manipulation Language)
  - b. Query
  - c. Relational Schema
  - d. DDL(Data Definition Language)
4. A Database Management System is a type of \_\_\_\_\_ software.
  - a. It is a type of system software
  - b. It is a kind of application software
  - c. It is a kind of general software
  - d. Both A and C
5. Which of the following refers to the level of data abstraction that describes exactly how the data actually stored?
  - a. Conceptual Level
  - b. Physical Level
  - c. File Level
  - d. Logical Level
6. In general, a file is basically a collection of all related \_\_\_\_\_
  - a. Rows & Columns
  - b. Fields
  - c. Database
  - d. Records
7. The term "Data" refers to:
  - a. The electronic representation of the information ( or data)
  - b. Basic information
  - c. Raw Facts and figures
  - d. Both A and C
8. The term "SQL" stands for
  - a. Standard query language
  - b. Sequential query language
  - c. Structured query language
  - d. Server-side query language
9. An \_\_\_\_\_ is a set of entities of the same type that share the same properties, or attributes.
  - a. Entity set
  - b. Attribute set
  - c. Relation set
  - d. Entity model
10. Which of the following gives a logical structure of the database graphically?
  - a. Database diagram
  - b. Entity diagram
  - c. Architectural representation
  - d. Entity-Relationship diagram
11. The entity relationship set is represented in E-R diagram as
  - a. Undivided rectangles
  - b. Dashed lines
  - c. Diamond
  - d. Double diamonds
12. In the \_\_\_\_\_ normal form, a composite attribute is converted to individual attributes.
  - a. First
  - b. Second
  - c. Third
  - d. Fourth
13. A table on the many side of a one to many or many to many relationship must:
  - a. Be in Second Normal Form (2NF)
  - b. Be in Third Normal Form (3NF)
  - c. Have a single attribute key
  - d. Have a composite key
14. Tables in second normal form (2NF):
  - a. Eliminate all hidden dependencies
  - b. Eliminate the possibility of a insertion anomalies
  - c. Have a composite key
  - d. Have all non key fields depend on the whole primary key
15. Which-one of the following statements about normal forms is FALSE?
  - a. BCNF is stricter than 3 NF
  - b. Lossless, dependency -preserving decomposition into 3 NF is always possible
  - c. Loss less, dependency – preserving decomposition into BCNF is always possible
  - d. Any relation with two attributes is BCNF
16. Functional Dependencies are the types of constraints that are based on \_\_\_\_\_
  - a. Key
  - b. Key revisited
  - c. Superset key
  - d. None of the mentioned

17. Which is a bottom-up approach to database design that design by examining the relationship between attributes:
- Functional dependency
  - Database modeling
  - Normalization
  - Decomposition
18. Which forms simplifies and ensures that there are minimal data aggregates and repetitive groups:
- 1NF
  - 2NF
  - 3NF
  - All of the mentioned
19. Which forms are based on the concept of functional dependency:
- 1NF
  - 2NF
  - 3NF
  - 4NF
20. A distributed database is a collection of data which belong \_\_\_\_\_ to the same system but are spread over the \_\_\_\_\_ of the network.
- Logically, sites
  - Physically, sites
  - Database, DBMS
  - None of the above
21. In a distributed Database reduction of redundancy is obtained by \_\_\_\_\_
- Data fragmentation
  - Data Replication
  - Data Sharing
  - None of the above
22. Which of the following is/are the main goals of a distributed database?
- Interconnection of database
  - Incremental growth
  - Reduced communication overhead
  - All of the above
23. The server process that implements lock manager functionality, is called
- Server Processes
  - Database Writer Process
  - Manager processes
  - Lock Manager Proceses
24. Storing a separate copy of the database at multiple locations is which of the following?
- Data Replication
  - Horizontal Partitioning
  - Vertical Partitioning
  - Vertical Partitioning
25. Which of the following is a group of one or more attributes that uniquely identifies a row?
- Key
  - Determinant
  - Tuple
  - Relation
26. A relationship is considered a:
- Column
  - one-dimensional table
  - two-dimensional table
  - three-dimensional table

### Answer Key

1.a	2.b	3.d	4.a	5.b	6.d	7.c	8.c	9.a	10.d
11.c	12.a	13.d	14.a	15.c	16.a	17.c	18.c	19.c	20.a
21.c	22.d	23.d	24.a	25.a	26.c				

## Exercise

### Short Answer Questions

- What are data, information and database?
- What is database? What are the objectives of a database?
- What is a database management system? How does database management system works?
- What is hierarchical data model?
- What is data normalization? What are the most important and widely used forms of data normalization?
- What is a relational database?
- What is a primary key? What are the features of good primary keys?

### ■ Long Answer Questions

1. What are the advantages and disadvantages of a database? Explain briefly.
2. Explain the difference between centralized database and distributed database.
3. What is data security? What are the different data security methods?
4. Why is normalization needed? Explain the normalization process with example.

### Project Work on DBMS Software

1. Make a case study to computerize the library system of your college. You can visit your library and ask the librarian what are the records they have to maintain. Then identify all the necessary entities and make an ERD and design the database in Microsoft Access with appropriate tables, queries, forms and reports.
2. Make a case study to computerize the billing system of a shopping center. You can visit the shopping center and ask them about the data they need to store. Then identify all the necessary entities and make an ERD and design the database in Microsoft Access with appropriate tables, queries, forms and reports.
3. Create a database named "School Database" and perform the following tasks
  - a. Create a table named "Students\_Info" with fields- Name, class, address, phoneNo section, rollNo and DOB
  - b. Fill at least 10 records
  - c. Make a query to display all records
4. Create a database named "library database" and perform the following tasks:
  - a. Create a table named "tbl\_Book" with the following structure:
  - b. Tbl\_Book(Book\_ID, Book\_Name, Author\_Name, Publication\_Name, Published\_year, Price, ISBN)
  - c. Add at least 10 records
  - d. Make a query to display all records where Author\_Name is equal to "Devendra"
  - e. Prepare a form on the basis of table
  - f. Prepare a report on the basis of above query.
5. Your school wants to maintain a real time database using Student admission record management system. Create a database to store the student's information and maintain their records.
6. Visit a Book store and make a case study to computerize the billing system of a book store. Identify all the entities and make DFD and design the database in Microsoft Access with appropriate table.
7. Design an ER Diagram for Library Management System and make a necessary database tables in Ms Access.
8. Design a database for Gym Management System

9. Design a database for online Quiz management system
10. Design a database for Bus Reservation system in MS Access
11. Design a database for Online Food Delivery System





# Concept of Network and Data Communication



## LEARNING OBJECTIVES

After going through this unit, you will be able :

- ◆ to learn basic elements of Communication System
- ◆ to know Concept of Communication System
- ◆ to draw Block Diagram of Communication System /Model
- ◆ to identify the Elements of Data Communication/Transmission
- ◆ to differentiate Simplex, Half duplex and Full duplex communication mode
- ◆ to learn Concept of LAN and WAN
- ◆ To know about Transmission Medium: Guided and Unguided
- ◆ to know Transmission impairments terminology
- ◆ to Basic concept of Networks Architecture: Client Server and Peer-to-peer
- ◆ to use Some Basic Terms and Tool Used in Computer Network
- ◆ to use Network Tool: Packet tracer, Remote Login
- ◆ to identify Network Connecting Devices: NIC, Modem, router, switch
- ◆ to know about Network Topologies: Bus, Ring and star topology
- ◆ to learn about Basic Concept OSI Reference Model
- ◆ to know Internet Protocol Addressing

## Basic Elements of Communication System

Data communications and networking is the fastest growing technologies in day-to-day life. Data communications and networking are changing the way we do business and the way we live. Businesses today rely on computer networks and internetworks.

Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable or wireless medium. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). There are three basic elements of communication system are:

- Transmitter
- Communication Channel
- Receiver

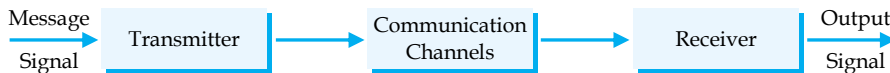


Figure: Communication System

The transmitter's function is to process the message signal into a form suitable for transmission over the communication channel. This is called modulation. As for the communication channel, its function is to provide a pathway between the transmitter's output and the receiver's input. The job of the receiver is to process the received signal to recover the appropriate message signal. If the different elements do their jobs accordingly, then the output signal should equal to the input message signal.

## Concept of Communication System

The communication system is a system model that describes a communication exchange between two stations, transmitter, and receiver. Signals or information passes from source to destination through a channel. It represents a way in which the signal uses it to move from a source toward its destination. To transmit signals in a communication system, it should first be processed by beginning from signal representation, to signal shaping until encoding and modulation. After the transmitted signal is prepared, it is passed to the transmission line of the channel. Due to signal crossing this media, it is faced with much impairment like noise, attenuation, and distortion. Some terms related to communication system

### Signal

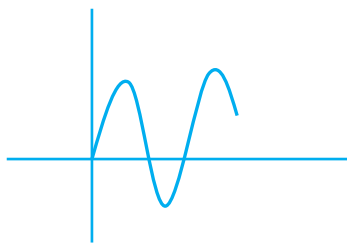
A signal is a medium for transmitting data or information. Signal can be of two types: Analog and Digital. Analog signals carry data in analog form and digital signal carry data in digital form.

In a network, a source encodes a message into a signal, which is carried to a receiver by the communications channel. For example, Sender wants to send a message "My name is Aayan"

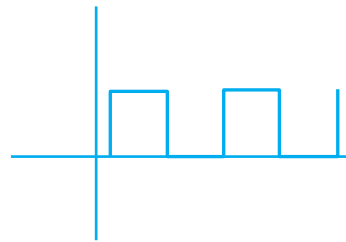
through a telephone line. The telephone converts the sound into electrical signals. The signal is now transmitted through telephone wires to the receiver. The receiver again converts the electrical signals into sound.

## Analog Signal

An analog signal is continuous and can be represented in sine waves. Human voice, listening music, television signal transmission are common examples of analog signals. Human voice generate analog (continuous varying) signal containing multiple frequencies that is transmitted as an analog signal over any medium. The signals which are vary in amplitude and frequency. Any information can be converted into an analog signal; such a signal is a measured response to changes in physical phenomena such as sound, light, temperature, position or pressure. The examples of analog signals are FM radio music, TV transmission, hearable audio, light etc.



Analog Signal



Digital Signal

## Digital Signal

A digital signal is a physical signal that is a representation of a sequence of discrete values. Digital signal describes any system based on discontinuous data or events. Computers are digital machines, at their most basic level they can distinguish between just two values 0 and 1 (OFF and ON case of electrical charge). Information transmitting in and out of the computer system is in the form of digital signals. The bandwidth of the digital transmission indicates how much data transfer in particular amount of time. Modern computer are digital machines because at their most basic level they can distinguish between just two values 0 and 1 or absence and presence of voltage. Hence, the data transmission between computer peripheral and inner parts of computer is digital transmission. For example: Pen drives to hard disk data transmission, DVD-ROM to Pen drive, RAM to Microprocessor are the common example of digital signal transmission.

There are some imperfections in transmission mediums. So, the signals sent through the transmission medium are also not perfect. This imperfection cause **signal impairment**. It means that signals that are transmitted at the beginning of the medium are not the same as the signals that are received at the end of the medium. What is sent is not what is received. The causes of signal impairment are:

### 1. Attenuation

Attenuation is the reduction in the strength of analog or digital signal as it is transmitted over a communication medium. It is a loss of energy of a signal as it travels. As the capacity of signal is reduced when it is transmitted over a long distance, amplifiers are used to boost the signal strength. The repeaters are used to regenerate the signal.

### 2. Distortion

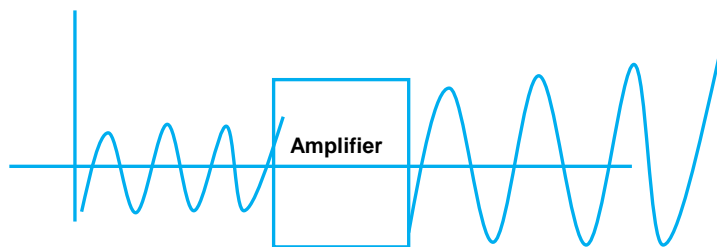
Distortion means that the signal changes its form or shape. Signals made up of different frequencies are composite signals. Distortion occurs in these composite signals.

### 3. Noise

Any electrical signal which interferes with an information signal is called noise. Also, noise is an error or undesired random disturbance of a useful information signal. Noise is most likely to be entered at the channel or the receiver. Example: Hiss in radio receivers, Buzz of telephone conversation, Flicker in television receivers, etc.

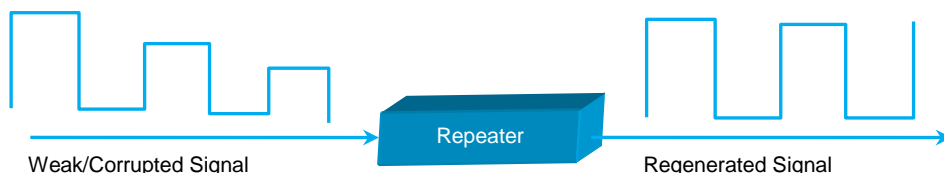
## Amplification

Amplification is the process to strengthen the amplitude of the signals using an electronic circuit. To amplify the signal amplifier are used.



## Repeater

The repeater extends the range of communication systems by re-generating the signals, allowing it to travel further. The signals can carry information within a fixed distance. After covering certain distance signal become corrupted or so weak. Repeaters help to regenerate the original signals and extend the length of travel.



## Modulation

As the original message signal can't be transmitted over an outsized distance due to their low frequency and amplitude, they're superimposed with high frequency and amplitude waves

called carrier waves. This phenomenon of superimposing a message signal with a carrier wave is called modulation. The process of changing or encoding the carrier wave at certain amplitude (height) and frequency (timing) is called modulation.

Some modulation technique change the height of the signal, some changes the timing of the signal and some change or revert the phase of the signal. In television transmission, the video signal is sent using amplitude modulation. The audio signal goes at a frequency modulation.

There are three different ways in which encoding of the analog signal with analog information is performed. Some of these methods are as follows:

- Amplitude Modulation (AM)
- Frequency Modulation (FM)
- Phase Modulation (PM)

### Amplitude Modulation (AM)

The amplitude (height) of the signal is changed depending on the information to be sent. Amplitude modulation is a technique used in electronic communication, most commonly for transmitting information via a radio carrier wave. Amplitude Modulation works by varying the strength of the transmitted signal in relation to the information being sent. For example, changes in the signal strength can be used to specify the sounds to be reproduced by a loudspeaker. This modulation techniques changes the amplitude of the signal wave, hence it is called amplitude modulation.

The modulation process is performed on the basis of original signal amplitude. If the amplitude of the signal is in positive (upper) portion, then the amplitude of modulated signal increases. Similarly, if the amplitude (height) of the signal is in negative (lower) portion then the amplitude of the modulated signal decreases. How much amplitude of the modulated signal increases and decreases depends on amplitude of the original signal.

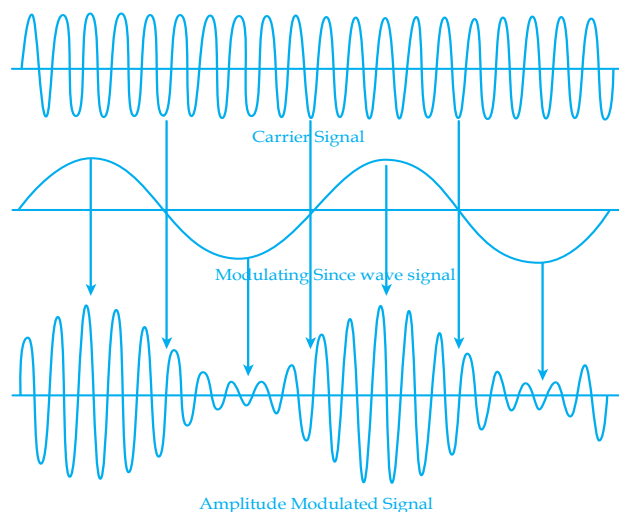


Figure: Amplitude Modulation (AM)

The AM transmission is not very good for transmitting high quality signal. The original signal can be altered by the atmosphere variables like lightning and quality degrades.

### Frequency Modulation (FM)

In contrast to amplitude modulation, frequency modulation conveys information over a carrier wave by changing the cycles of frequency. The amplitude and phase of the carrier signal are not changed at all. This method has less amount of noise and gives the best encoding method as far as quality is concerned.

The modulation process is performed on the basis of original signal frequency. If the frequency of the signal is in positive (upper) portion, then the frequency of modulated signal increases. Similarly, if the amplitude (height) of the signal is in negative (lower) portion then the frequency of the modulated signal decreases. How much frequency of the modulated signal increases and decreases depends on amplitude of the original signal.

The FM transmission is very good for transmitting high quality signal. The original signal cannot be altered by the atmosphere variables and hence the quality remains same.

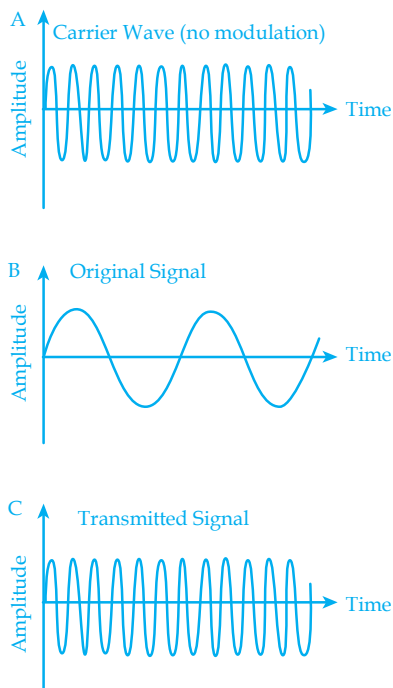


Figure: Frequency Modulation (FM)

A good example of such encoding is FM broadcasting. At first, digital sound (CD audio) is converted into analog electrical signal. The analog signal is encoded in the base of frequency and transmitted is being done. The reverse process is performed at the FM radio set to recover the information in electrical form, and we hear what was being transmitted from the FM station.

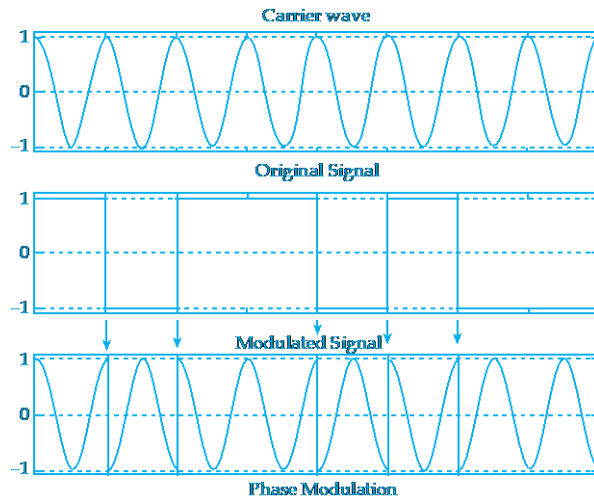


Figure: Phase Modulation (PM)

### Phase Modulation (PM)

In phase modulation, the amplitude and frequency of the carrier signal remains unchanged but alters its phases. The characteristics of this encoding technique are similar to FM. Phase modulation is a form of modulation that can be used for radio signals used for a variety of radio communications applications. As will be seen later, phase modulation, and frequency modulation are closely linked together and it is often used in many transmitters and receivers used for a variety of radio communications applications from two way radio communications links, mobile radio communications and even maritime mobile radio communications.

Phase modulation is also the basis for many forms of digital modulation based around phase shift keying (PSK), which is a form of phase modulation.

### Demodulation

Demodulation takes a modulated signal and then extracts the original message from it. Demodulator is used to recover the information content from the modulated carrier wave. Demodulation is the reverse process of modulation. Modulating a signal is useful for long distance transmission. At the same time, demodulation is also necessary to recover the modulated signals. A modem is a device that is used for modulation and demodulation both.

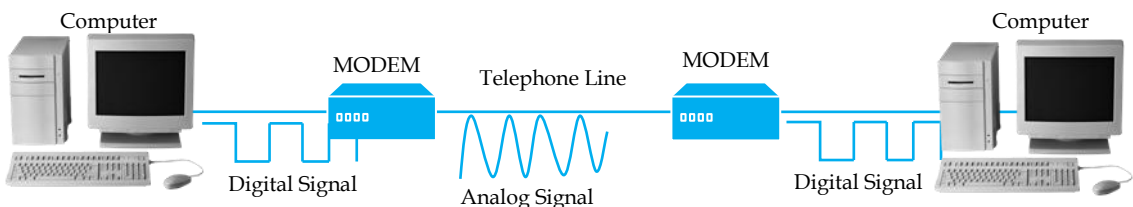


Figure: Modulation/ Demodulation

Conversion of analog signal is known as Modulation; conversion back to digital (binary signals) is known as demodulation. MODEM was initially used for connecting users to the internet but today MODEM can be used in various fields. Some of the application of MODEM includes:

- Point of Sales(POS)
- Home Security Monitoring
- Data Transfers
- Broadband Backup
- Industrial Control Applications
- Remote Management

## Communication System /Model

Communication system is a system model that describes a communication exchanges between two stations, transmitter and receiver. Signals or message passes from source to destination through channel, which represents a way that signal use it to move from source toward destination.

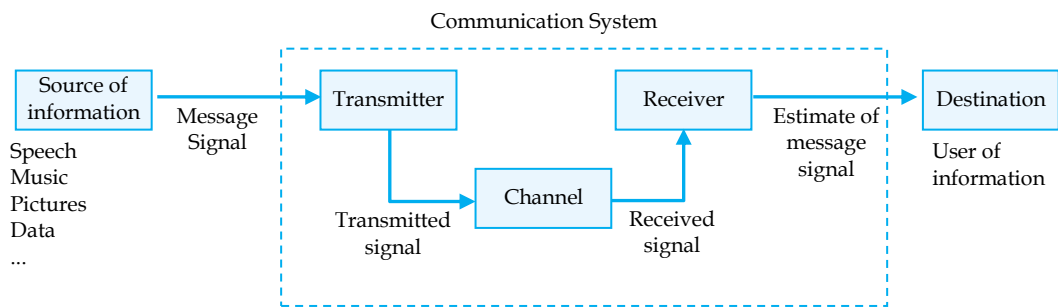


Figure: Block Diagram of Communication system

The purpose of a communication system is to carry information from one point to another. A typical communication system consists of three main components as shown in figure below: source, channel and destination. Source generates information and Destination reads the information which passes through channel from source to destination. The communication channel can be physical/wired such as twisted pair cable, coaxial cable or logical/wireless such as radio wave depending upon the distance of data to be travel. A channel has certain capacity for transmitting information, often measured by its bandwidth in Hz. The block diagram of communication system is given below:

## Elements of Data Communication/Transmission

There are five elements of data communication/transmission. They are:

- Message
- Sender

- Receiver
- Transmission Medium and
- Protocol

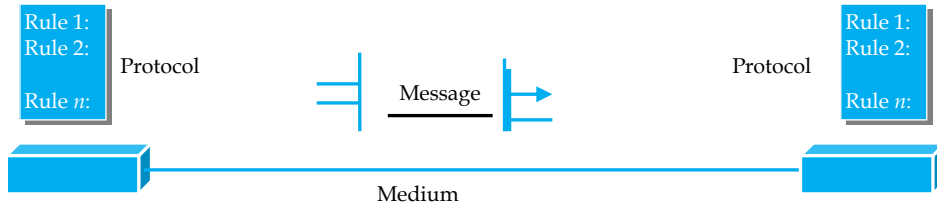


Figure: Elements of data communication

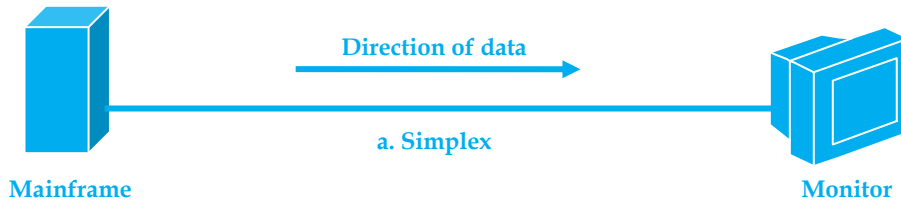
1. **Message:** The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
2. **Sender:** The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.
3. **Receiver:** The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
4. **Transmission Medium:** The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.
5. **Protocol:** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

## Communication Mode

Communication mode refers to the mechanism of transferring of data between two devices connected over a network. It is also called Transmission Mode. These modes direct the direction of flow of information. Communication between two devices can be simplex, half-duplex or full duplex.

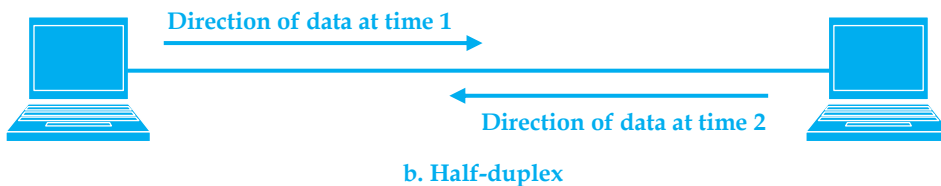
### 1. Simplex Mode

In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive. Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output. The simplex mode can use the entire capacity of the channel to send data in one direction.



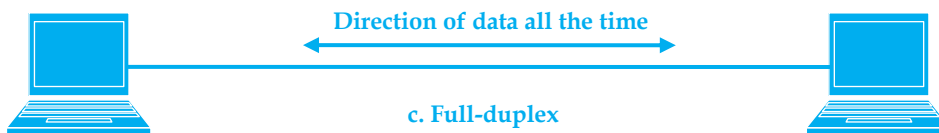
## 2 Half-Duplex Mode

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa. Example: Walkie-talkie. The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.



## 3. Full-Duplex Mode

In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously. The full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the two directions. Example: Telephone network.



## Concept of LAN and WAN

A network is a set of devices (often referred to as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network. The network categories into two primary categories:

- Local- Area Network (LAN) and
- Wide-Area Network (WAN)

### 1. Local-Area Network (LAN)

A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus. Depending on the needs of an organization and the type of technology used, a LAN can be as simple as two PCs and a printer in someone's home office; or it can extend throughout a company and include audio and video peripherals. Currently, LAN size is limited to a few kilometers.



Figure: Local Area Network (LAN)

LANs are designed to allow resources to be shared between personal computers or workstations. The resources to be shared can include hardware (e.g., a printer), software (e.g., an application program), or data. LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium. The most common LAN topologies are bus, ring, and star.

Early LANs had data rates in the 4 to 16 megabits per second (Mbps) range. Today, however, speeds are normally 100 or 1000 Mbps. Wireless LANs are the newest evolution in LAN technology.

#### Example:

Engineering workstations or accounting PCs. One of the computers may be given a large capacity disk drive and may become a server to clients. Software can be stored on this central server and used as needed by the whole group. In this example, the size of the LAN may be determined by licensing restrictions on the number of users per copy of software, or by restrictions on the number of users licensed to access the operating system.

#### Advantages

- **Resource Sharing:** LAN provides resource sharing such as computer resources like printers, scanners, modems, DVD-ROM drives, and hard disks within the connected devices which reduce cost and hardware purchases.
- **Easy and Cheap Communication:** Data and messages can easily be shared with the other computer connected to the network.
- **Centralized Data:** The data of all network users can be stored on a hard disk of the central/server computer. These help users to use any computer in a network to access the required data.
- **Data Security:** Since data is stored on the server computer, it will be easy to manage data at only one place and the data will be more secure too.

### Disadvantages

- **High Setup Cost:** The initial setup costs of installing Local Area Networks is high because there is special software required to make a server. Also, communication devices like an Ethernet cable, switches, hubs, routers, cables are costly.
- **Data Security Threat:** Unauthorized users can access important data of an office or campus if a server hard disk is not properly secured by the LAN administrator.
- **LAN Maintenance Job:** Local Area Network requires a LAN Administrator because there are problems such as software installations, program faults or hardware failures or cable disturbances in Local Area Network. A LAN Administrator is required to maintain these issues.
- **Covers Limited Area:** LANs are restricted in size they cover a small area like a single office, single building or a group of nearby buildings.

## 2. Wide-Area Network (WAN)

A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world. A WAN can be as complex as the backbones that connect the Internet or as simple as a dial-up line that connects a home computer to the Internet. The switched WAN connects the end systems, which usually comprise a router (internetworking connecting device) that connects to another LAN or WAN. The point-to-point WAN is normally a line leased from a telephone or cable TV provider that connects a home computer or a small LAN to an Internet service provider (ISP). This type of WAN is often used to provide Internet access.

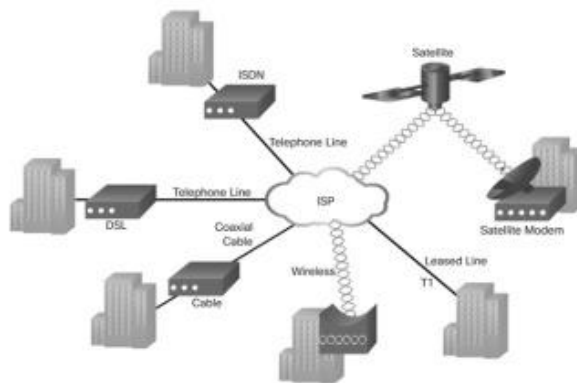


Figure: Wide-Area Network

### Advantages

- Covers large geographical area
- Centralized data

Wide area networks also provide you the facility of sharing the data to all of your connected devices in a network.

- Get updated files and data

Wide Area Networks provide the facility of getting updated files and data from the server. If a server is updated with new data, then all connecting devices receive that updated data within seconds.

- Sharing of software and resources

Like LANs, WANs share software applications and other resources with other users on the internet.

- High bandwidth

WAN covers a large geographical area therefore WANs have high bandwidth compared to LANs and MANs.

### Disadvantages

- Security problems

Wide Area Networks faces more security problem as compare to LANs and MANs.

- Needs firewall and antivirus software

As it faces security issue, therefore it is a basic need of WANs to use firewalls and antivirus software to protect data.

- The setup cost is high

A WAN network covers a large geographical area; it is very expensive to setup in the initial stage.

- Troubleshooting problems

A WAN network covers large geographical areas, so fixing the problem in a network is a very difficult job.

- Maintenance Issues

Once set up, maintaining a WAN network is a full-time job which requires high tech skills of network supervisors and technicians.

### Metropolitan Area Networks (MAN)

A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high-speed ADSL line to the customer.

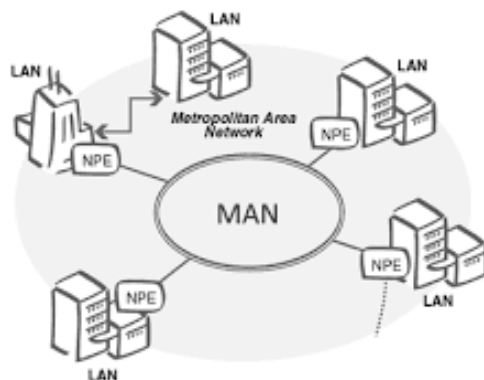


Figure: Metropolitan Area Networks (MAN)

### Advantages

- High Speed compared to WAN
- High Security – It provides high security compared to WAN and LAN.
- Not very expensive

### Disadvantages

- Difference in Internet Speed
- Security problems
- Maintenance Issues

## Difference between Local Area Network (LAN) and Wide Area Network (WAN)

Basis of Comparison	Local Area Network	Wide Area Network
Definition	LAN is a computer network that covers a small geographic area, like a home, office, or group of buildings.	WAN is a computer network that covers a large geographic area likes cross-regional and metropolitan boundaries over a long distance.
Speed	LAN speed is quiet high.	WAN speed is lower than that of LAN.
Congestion	LAN has low congestion as compared to WAN.	WAN has higher congestion than both MAN and LAN.
Fault Tolerance	Fault Tolerance of LAN is higher than WAN.	Fault Tolerance of WAN is lower than both LAN and MAN.
Maintenance	Designing and maintaining LAN is easy and less costly than WAN.	Designing and maintaining WAN is complex and more costly than both LAN and MAN.

## Transmission Medium: Guided and Unguided

Transmission medium is a path through which information can be transmit from a sender to receiver. The design factors of Transmission Medium are: Bandwidth, Transmission impairments, Interference and Number of receivers. There are two main categories of transmission medium:

- Guided Transmission Medium
- Unguided Transmission Medium

### 1. Guided Transmission Medium

Guided transmission media are also called bounded media or wired media. Guided medium provide conduit from one device to another, the signal is directed and contained by the physical limits of the medium. Guided medium includes:

- Twisted Pair Cable:** It is the oldest, least expensive and most commonly used media. Pair of insulated wires twisted together to reduce susceptibility to interference. It is quite high susceptible to noise and interference. Twisted pair cable needs repeater for every 2-3 km for digital transmission of data and amplifier for every 5-6 km for analog data. It comes in two forms:
  - **Unshielded Twisted Pair (UTP):** Most common type in use today. The plastic insulation is color-banded for identification. Significantly reduced electromagnetic noise interference compared with two parallel flat wires. It is Cheap, flexible, and easy to install.
  - **Shielded Twisted Pair (STP):** It has a metal foil or braided-mesh covering that encases each pair of insulated conductors. The metal casing prevents the penetration of electromagnetic noise. It eliminates most crosstalk.



Figure: Twisted Pair Cable

- Coaxial Cable:** The coaxial cables have a central copper conductor, surrounded by an insulating layer, a conducting shield, and the outermost plastic sheath. Thus, there are three insulation layers for the inner copper cable. There are two basic modes of data

transmission in coaxial cables: baseband mode that has dedicated bandwidth, and broadband mode that has distributed cable bandwidth. Coaxial cables have better resistance to cross talk than twisted pair cables. The coaxial cables are used for long distance communication.

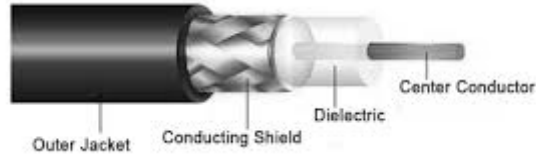


Figure: Coaxial Cable

- c. **Fiber Optics:** Optical fiber is flexible, thin (few to few hundred  $\mu\text{m}$ ), very pure glass / plastic fiber capable of conducting optical rays. Optical fibers use light waves for transmission. The fiber optical cable uses the application of total internal reflection of light. The fibers are designed such that they facilitate the propagation of light along with the optical fiber depending on the requirement of power and distance of transmission. Single-mode fiber is used for long-distance transmission, while multimode fiber is used for shorter distances. The outer cladding of these fibers needs better protection than metal wires. It is especially use for long-distance communications. It has extremely high bandwidth.

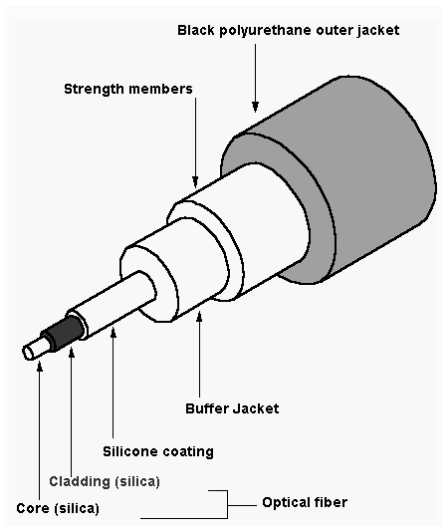


Figure: Optical Fiber

## Difference between UTP and STP

Basis of Comparison	Unshielded Twisted Pair (UTP)	Shielded Twisted Pair (STP)
Electromagnetic Interference	Electromagnetic interference and noise is more in UTP.	STP cable reduce electrical noise within the cable and from outside of the cable.
Speed	It offers speed or throughput of about 10 to 1000Mbps.	It offers speed or throughput of about 10 to 100 Mbps.
Distance	It offers maximum cable length of about 100 meters.	It supports maximum segment of length about 100 meters.
Application	UTP is widely used for data transmission within short distance and is very popular for home network connecting.	STP is mainly used for connection of enterprises over a long distance.
Cost	The cost of UTP is less when compared to that of STP.	STP is costlier than UTP.

## 2. Unguided Transmission Medium

It is also referred to as Wireless or Unbounded transmission media. No physical medium is required for the transmission of electromagnetic signals. Uses the section of electromagnetic spectrum known as radio communication (3 KHz - 300 GHz), which is divided into eight bands, to transmit signals. Unguided medium is:

- a. **Microwave:** It is typically used where laying a cable is not practical. Parabolic dish shaped antenna (≈10 ft diameter). It transmits/receives electromagnetic waves in the 2-40 GHz range. Microwave travels in a straight line (line-of-sight propagation). The high rate of data transmission is 100 Mbps. The repeater should be placed at 10 - 100 km apart. It is normally used in long distance telephone communication.



Figure: Microwave

- b. **Satellite:** It uses satellite in geostationary (geosynchronous) orbit ( $\approx 36,000$  km). The source transmits signal to satellite which amplifies or repeats it, and retransmits down to destinations. The optimum transmission is in the range of 1 -10 GHz and bandwidth of 100's MHz. The propagation delay is nearly 270ms. The total propagation delay is independent of distance between sender and receiver. The application of satellite is in long distance telephone, television distributions, etc.

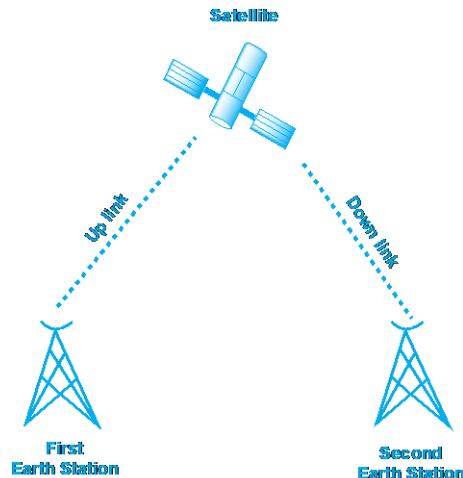


Figure: Satellite Communication

- c. **Infrared Signal:** Infrared waves, with frequencies from 300 GHz to 400 THz, can be used for short-range communication like TV remote, wireless speakers, automatic doors, hand held devices etc. Infrared signals can propagate within a room but cannot penetrate walls. However, due to such short range, it is considered to be one of the most secure transmission modes.

### Difference between Guided and Unguided Transmission Medium

Basis of Comparison	Guided Transmission Medium	Unguided Transmission Medium
Definition	Guided medium provide conduit from one device to another, the signal is directed and contained by the physical limits of the medium.	Unguided medium has no physical medium for the transmission of electromagnetic signals.
Alternative Name	It is called wired communication or bounded transmission media.	It is called wireless communication or unbounded transmission media.
Signal Propagation	The signal energy propagates through wires in guided media.	The signal energy propagates through air in unguided media.
Types	Twisted pair cable, coaxial cable and fiber optic cable.	Radio wave, microwave and infrared.
Example	Twisted pair wires, coaxial cables and optical fiber cables.	Microwave or radio links and infrared light.

## Transmission Impairments Terminology

Transmission impairment means the signal corruption during transmission. The transmission impairment is:

### Jitter

Jitter is a problem if different packets of data encounter different delays and the application using the data at the receiver site is time-sensitive such as audio and video data. If the delay for the first packet is 20 ms, for the second is 45 ms, and for the third is 40 ms, then the real-time application that uses the packets endures jitter.

### Singing

Singing occurs as a result of transmitted signals being coupled into a return path and fed back to the respective sources. It is the same as echo.

### Echo

Echo occurs when some portion of the signal is returned/arrived with enough delay (typically more than about 30 ms) to make the signal distinguishable from normal side tone. Echo are of two types: Talker Echo and Listener Echo. Talker echo occurs when some portion of the talker's speech signal is returned. Talker echo may be caused by reflections at impedance mismatches or by other processes such as go-to-return crosstalk. The effect of talker echo is a function of the loss in the acoustic-to-acoustic echo path and the delay in the echo path. Listener echo refers to a transmission condition in which the main speech signal arrives at the listener's end of the connection accompanied by one or more delayed versions (echoes) of the signal. Such a condition can occur as the result of multiple reflections in the transmission path.

### Crosstalk

Crosstalk is the effect of one wire on the other. One wire acts as a sending antenna and the other as the receiving antenna. Crosstalk may affect both wires and create unwanted signals. We can normally say crosstalk as unwanted coupling between signal paths.

### Distortion

Distortion is any changes in a signal due to noise, attenuation or other influences. Distortion can occur in a composite signal made of different frequencies. Each signal component has its own propagation speed through a medium and, therefore, its own delay in arriving at the final destination. Differences in delay may create a difference in phase if the delay is not exactly the same as the period duration.

### Noise

Noise is another cause of impairment. Several types of noise, such as thermal noise, induced noise, crosstalk, and impulse noise, may corrupt the signal. Thermal noise is the random motion of electrons in a wire which creates an extra signal not originally sent by the transmitter.

Induced noise comes from sources such as motors and appliances. These devices act as a sending antenna, and the transmission medium acts as the receiving antenna. Crosstalk is the effect of one wire on the other.

### Bandwidth

The rate at which digital data can be transmitted over a given communication channel. The bandwidth of the transmitted signal as constrained by the transmitter and the nature of the transmission medium, expressed in cycles per second, or Hertz.

### Number of Receivers

As the number of receivers increases, the rate of data transmission decreases.

## Basic Concept of Networks Architecture: Client Server and Peer-to-peer

Network architecture is the logical and structural layout of the network, consisting of transmission equipment, software and communication protocols, and infrastructure (i.e. wired or wireless) transmission of data and connectivity between components. There are two types of network architecture is: Client Server and Peer-to-peer.

### Client-Server Architecture

Client-server architecture is a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). In a client/server network, a centralized, really powerful computer (server) acts as a hub in which other computers or workstations (clients) can connect to. This server is the heart of the system, which manages and provides resources to any client that requests them.

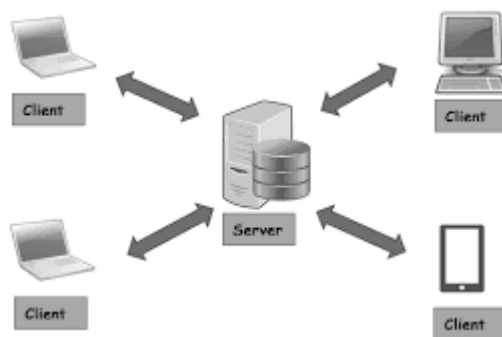


Figure: Client-Server Architecture

### Advantages

- Resources and data security are controlled through the server.
- Not restricted to a small number of computers.

- Server can be accessed anywhere and across multiple platforms.
- It has centralized control
- It can reduce data replication
- Due to centralized architecture data is secure

### Disadvantages

- Can become very costly due to the need of a server as well as networking devices such as hubs, routers, and switches.
- If the server goes down, the entire network will be affected.
- Technical staff needed to maintain and ensure network functions efficiently.
- It cannot handle too many client requests. Server will be overloaded and can create problems in accessing information.

### Peer-to-Peer Architecture

In a peer-to-peer network, tasks are allocated to every device on the network. Furthermore, there is no real hierarchy in this network, all computers are considered equal and all have the same abilities to use the resources available on this network. Instead of having a central server which would act as the shared drive, each computer that's connected to this network act as the server for the files stored on it.

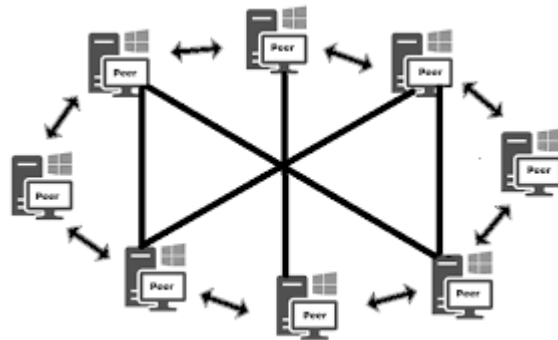


Figure: Peer-to-Peer Architecture

### Advantages

- Does not require a dedicated server which means it's less costly.
- If one computer stops working, the other computers connected to the network will continue working.
- Installation and setup is quite painless because of the built-in support in modern operating systems.
- The files or resources are equally shared among all the users.
- It is scalable. It can handle large number of users without degrading its performance.

### Disadvantages

- Security and data backups are to be done to each individual computer.
- As the numbers of computers increases on a P2P network, security, and access becomes a major issue.
- The files are not centrally organized in p2p Network.
- Since data is not centralized, data backup is more difficult.
- Peer to peer networks are more prone to malware and virus attacks

### Difference between Client-Server Architecture and Peer-to-Peer Architecture

Basis of Comparison	Client-Server Architecture	Peer-to-Peer Architecture
Focus	Focuses on information sharing.	Focuses on connectivity.
Data	Centralized server is used to store the data.	Each peer has its own data.
Stability	Client-Server Network is more stable and scalable.	Peer-to-Peer Network are less stable if number of peer is increased.
Reliability	It is less reliable.	It is more reliable.
Security	It is very much secure when compared the Peer-to-Peer Network. This is because the server can authenticate a client's access to any part of the network.	It is less secure when compared to Client-server Network and in fact it becomes more problematic as the number of clients increases.
Access Time	Multiple clients request for services from a server and therefore, the access time for a service is higher.	The service providing nodes are distributed, the service requesting node does need to wait long.

### Some Basic Terms and Tool Used in Computer Network:

#### IP Address

An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network. An IP address is set of unique 8-bit numbers assigned to a device that connect to a network. IP addresses are the identifier that allows information to be sent between devices on a network: they contain location information and make devices accessible for communication.

There are two IP versions: IPv4 and IPv6. IPv4 is the older version which has a space of over 4 billion IP addresses. However, the new IPv6 version can provide up to trillions of IP addresses to fulfill the need of all internet users and devices. Example: 172.16.254.1 in IPv4, and 2001:db8:0:1234:0:567:8:1 in IPv6.

## Need of IPV6

The numbers of internet based devices are developing as such a fast speed so that the possible number of devices that can be connected using IPV4 may lack soon. IPV4 uses 32 bit address (It can connect  $2^{32}$  unique devices at a time) but IPV6 uses 128 bit address (It can support  $2^{128}$  unique devices at a time) which is capable of providing huge number of different IP addresses which can easily fulfill the future demand. The advantage of IPV6 over IPV4 includes:

- The IPV6 address is 1028 times larger than IPV4 address
- Auto-configuration
- Better multicast routing
- Simpler header format
- Easier administration
- Flexible extensions
- Improved Performance and Security

## Sub Net Mask and Gateway

Subnet is a logical subdivision of an IP network. The practice of dividing a network into two or more networks is called subnetting. Subnetting increases routing efficiency, which helps to enhance the security of the network and reduces the size of the broadcast domain.

A subnet mask is a 32 bits' address used to distinguish between a network address and a host address in IP address. A subnet mask identifies which part of an IP address is the network address and the host address. They are not shown inside the data packets traversing the Internet. They carry the destination IP address, which a router will match with a subnet. There are two types of subnet masks: The default Subnet Mask, which is the number of bits which is reserved by the address class and A Custom Subnet Mask, which can be defined by an administrator to accommodate many Networks.

A gateway is normally a computer that operates in all five layers of the Internet or seven layers of OSI model. A gateway takes an application message, reads it, and interprets it. This means that it can be used as a connecting device between two internetworks that use different models. For example, a network designed to use the OSI model can be connected to another network using the Internet model. The gateway connecting the two systems can take a frame as it arrives from the first system, move it up to the OSI application layer, and remove the message.

## MAC Address

A MAC address, or Media Access Control address, is a 48- or 64-bit address associated with a network adapter. While IP addresses are associated with software, MAC addresses are linked to the hardware of network adapters. For this reason, the MAC address is sometimes called the hardware address, the burned-in address (BIA), or the physical address.

MAC addresses are expressed in hexadecimal notation in the following format: 01:23:45:67:89: AB, in the case of a 48-bit address or 01:23:45:67:89: AB: CD: EF, in the case of a 64-bit address. Dash (-) are sometimes used instead of colon (:). MAC addresses are often considered permanent, but in some circumstances, they can be changed. There are two types of MAC addresses: Universally Administered Address (UAA) is the most commonly used type of MAC address. This address is assigned to the network adapter when it is manufactured. The first three octets define the manufacturer, while the second three octets vary and identify the individual adapter. All network adapter manufacturers have their own code, called the Organizationally Unique Identifier (OUI). And Locally Administered Address (LAA), is an address that changes an adapter's MAC address. The LAA is a type of administered MAC address, and it is possible to change the LAA of a network adapter to any address of allowed length. When the LAA is set, the network adapter uses the LAA as its MAC address.

### Internet

An internet is the global system of interconnected computer networks to communicate between networks and devices. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing.

The concept of Internet was originated in 1969 and has undergone several technological & Infrastructural changes as discussed below: The origin of Internet devised from the concept of Advanced Research Project Agency Network (ARPANET). ARPANET was developed by United States Department of Defense. Basic purpose of ARPANET was to provide communication among the various bodies of government. Initially, there were only four nodes, formally called Hosts. In 1972, the ARPANET spread over the globe with 23 nodes located at different countries and thus became known as Internet. By the time, with invention of new technologies such as TCP/IP protocols, DNS, WWW, browsers, scripting languages etc., Internet provided a medium to publish and access information over the web.

### Intranet

An intranet is a private network (LAN) that uses the Internet model. However, access to the network is limited to the users inside the organization. The network uses application programs defined for the global Internet, such as HTTP, and may have Web servers, print servers, file servers, and so on.

### Extranet

An extranet is the same as an intranet with one major difference: Some resources may be accessed by specific groups of users outside the organization under the control of the

network administrator. For example, an organization may allow authorized customers access to product specifications, availability, and online ordering. A university or a college can allow distance learning student's access to the computer lab after passwords have been checked.

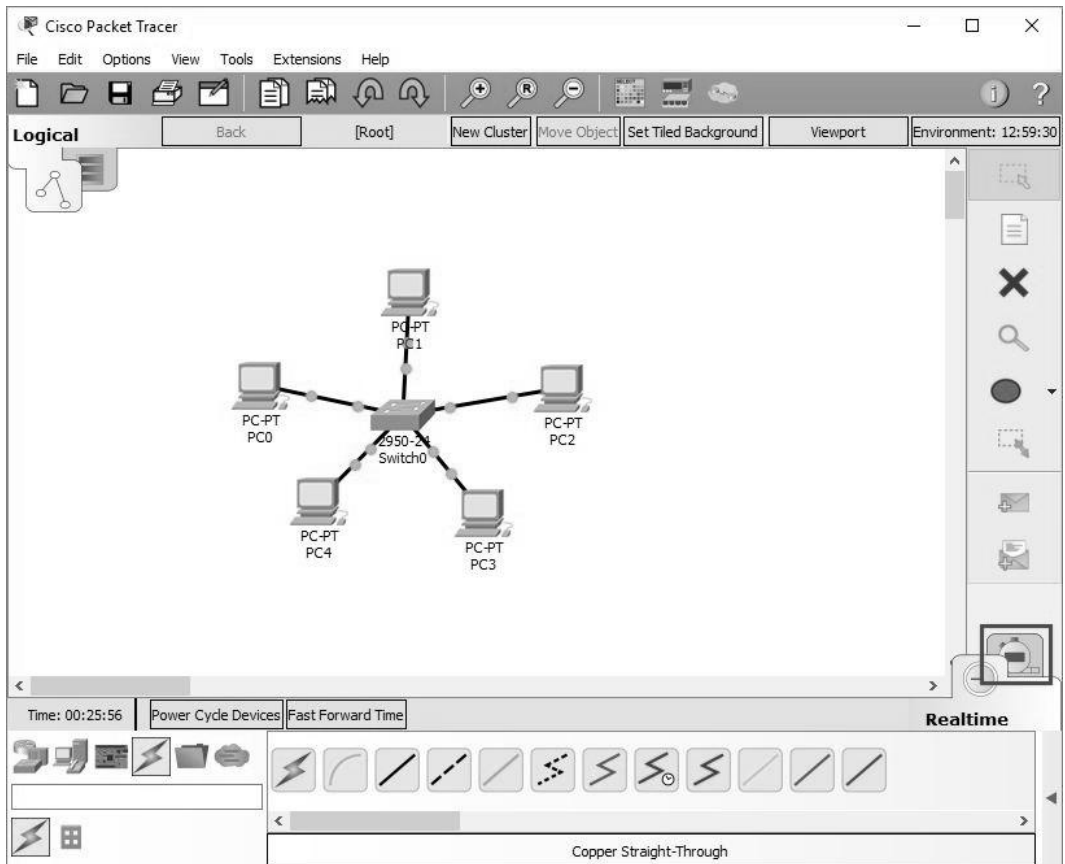
## Network Tool: Packet tracer, Remote Login

### Packet Tracer

Packet Tracer is a comprehensive networking technology teaching and learning tool that offers a unique combination of realistic simulation and visualization experiences, assessment, activity authoring capabilities, and multiuser collaboration and competition opportunities. Innovative features of Packet Tracer will help students and teachers collaborate, solve problems, and learn concepts in an engaging and dynamic social environment.

### Advantages

- Provides a realistic simulation and visualization learning environment that supplements classroom equipment, including the ability to see internal processes in real-time that are normally hidden on real devices
- Enables multi-user, real-time collaboration and competition for dynamic learning
- Enables authoring and localization of structured learning activities such as labs, demonstrations, quizzes, exams, and games
- Empowers students to explore concepts, conduct experiments, and test their understanding of network building
- Allows students and teachers to design, build, configure, and troubleshoot complex networks using virtual equipment
- Supports a variety of teaching and learning opportunities such as lectures, group, and individual labs, homework, games, and competitions
- Supports feature expansion through external applications using an API to enhance Cisco Packet Tracer's functionality in areas such as curriculum and assessment delivery, games, accessibility, and interfacing with real equipment.



## Remote Login

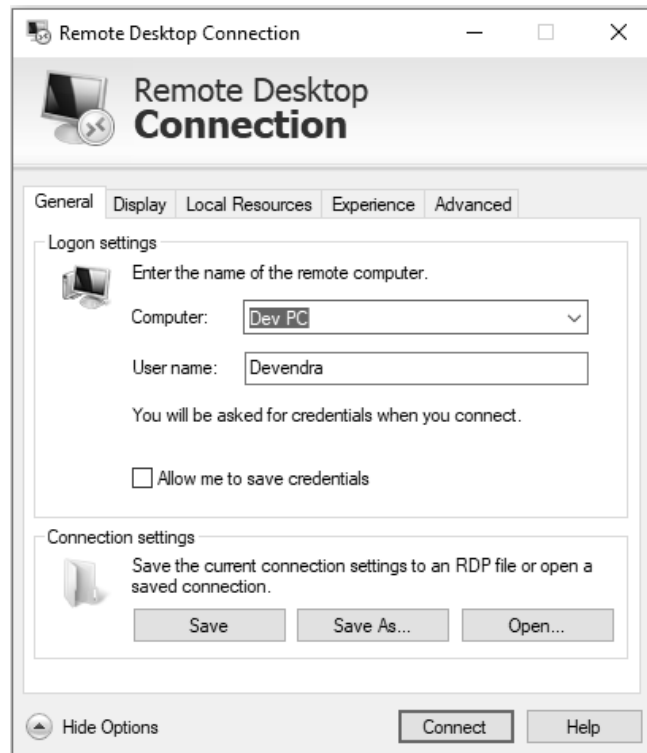
A login that allows a user terminal to connect to a host computer via a network or direct telecommunications link and to interact with that host computer as if the user terminal were directly connected to that host computer is remote login.

A remote desktop connection allows you to use virtually any computer or device in order to connect to another computer. Once you connect remotely, you get access to all the resources on the computer you've connected to. That means you get control over the mouse and the keyboard and you can see everything happening on the screen of the computer you've connected to.

Remote login may be very useful in some situation like: imagine a situation when you have to meet a tight deadline and you can't do all the work at your office, but you need to work on your office computer. Wouldn't it be great if you could stay at your home but still work on your office computer? That is, without physically taking your office computer with you. That is possible if you set a remote desktop connection between your work computer and your home computer. The one in your office would be the host (because you're working on it) and the

computer at home would be the client (as it's the one used to establish the connection). Then, after the remote desktop connection is set, you could sit in front of your home computer but you'd actually be working on your office computer.

Most of the recent version of windows supports remote desktop connection. You have to setup the connection before you share the files. The remote desktop connection on windows looks like:



## Network Connecting Devices: NIC, Modem, Router, Switch

### NIC (Network Interface Card)

An electronic device, internal or external to a station, that contains circuitry to enable the station to be connected to the network. It is a circuit board installed in a computer that provides a dedicated network connection to the computer. Early network interface controllers were commonly implemented on expansion cards that plugged into a computer bus. Modern network interface controllers offer advanced features such as interrupt and DMA interfaces to the host processors, support for multiple receive and transmit queues, partitioning into multiple logical interfaces, and on-controller network traffic processing such as the TCP offload engine.



Figure: NIC Card

## Modem

The term modem is a composite word that refers to the two functional entities that make up the device: a signal modulator and a signal demodulator. A modulator creates a bandpass analog signal from binary data. A demodulator recovers the binary data from the modulated signal. Most stand-alone modems have just two ports: one that connects to the outside world, and an Ethernet jack that connects to a computer or a router.



Figure: MODEM

## Router

A router is a three-layer device that routes packets based on their logical addresses (host-to-host addressing). A router normally connects LANs and WANs in the Internet and has a routing table that is used for making decisions about the route. The routing tables are normally dynamic and are updated using routing protocols.



Figure: Router

## Switch

Switch is a bridge, a bridge with many ports and a design that allows better (faster) performance. A switch, as a bridge does, makes a filtering decision based on the MAC address of the frame it received. However, a switch can be more sophisticated. It can have a buffer to hold the frames for processing. It can have a switching factor that forwards the frames faster. Some new switches, called cut-through switches, have been designed to forward the frame as soon as they check the MAC addresses in the header of the frame.



Figure: Switch

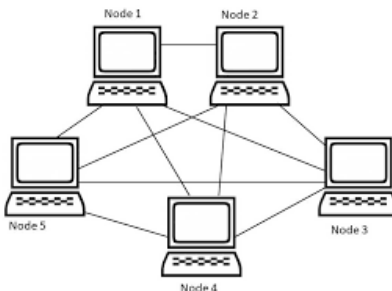
## Network Topologies: Bus, Ring and Star Topology

The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. Topology refers to the way in which the networking devices are physically connected. The common basic types of topologies are:

- Bus topology
- Star topology
- Tree topology
- Ring topology
- Hybrid Topology
- Mesh topology

### Mesh Topology

In a mesh topology, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects. To find the number of physical links in a fully connected mesh network with  $n$  nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to  $n - 1$  nodes, node 2 must be connected to  $n - 1$  nodes, and finally node  $n$  must be connected to  $n - 1$  nodes. We need  $n(n - 1)$  physical links. However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2.



### Advantages

- The use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices.
- A mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system.
- Privacy and security
- Point-to-point links make fault identification and fault isolation easy.

### Disadvantages

- Need more cable and ports.
- The sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate. The hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.

### Star Topology

In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub/switch. The devices are not directly linked to one another. Unlike a mesh topology, a star topology does not allow direct traffic between devices. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device.

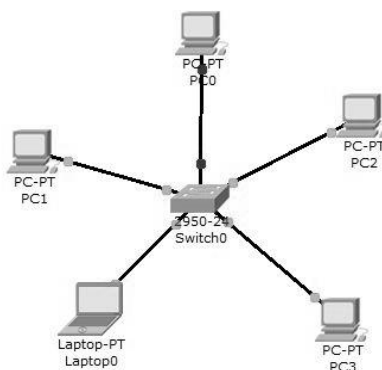


Figure: Star topology

### Advantages

- It is less expensive than Mesh topology.
- Robustness: If one link fails, only that link is affected. All other links remain active.
- Easy fault identification and fault isolation.
- As long as the hub is working, it can be used to monitor link problems and bypass defective links.

### Disadvantages

- Dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead.
- Although a star requires far less cable than a mesh, each node must be linked to a central hub. For this reason, often more cabling is required in a star than in some other topologies (such as ring or bus).
- Data traffic, data collision and data error rate will be high.

### Tree Topology

A tree topology is the extension of star topology. A tree topology includes multiple star topologies, which involve a variety of single nodes connected to a central node. The main advantages of this topology are these are very flexible and also have better scalability. Tree topology is one of the most popular among network topology.

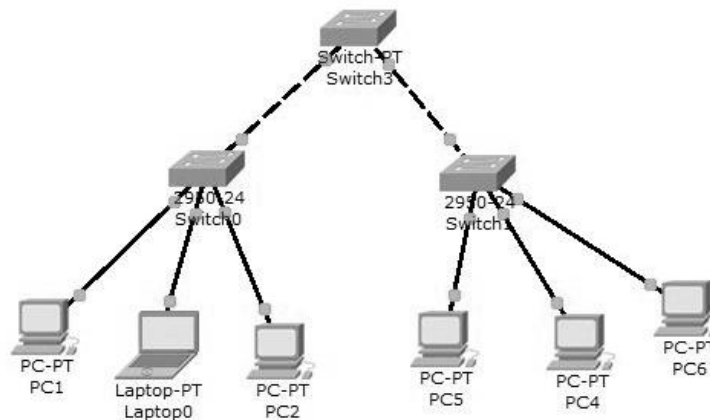


Figure: Tree topology

### Advantages

- Scalable as leaf nodes can accommodate more nodes in the hierarchical chain.
- Other hierarchical networks are not affected if one of them gets damaged
- Easier maintenance and fault finding

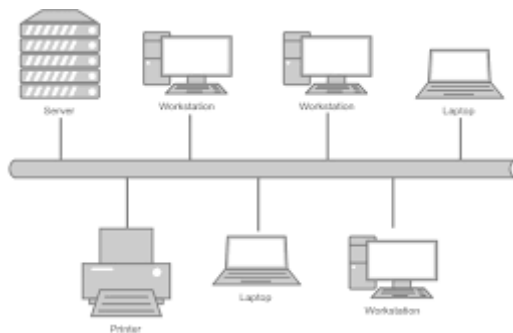
- Expansion of network is possible and easy.
- Suitable of large organizations having large number of computers.
- If one segment is damaged other is not affected.

### Disadvantages

- Huge cabling is needed
- A lot of maintenance is needed
- As number of computer increases maintenance become difficult.
- If main switch fails, entire network fails
- Cost of establishing the topology increases

### Bus Topology

A bus topology, on the other hand, is multipoint. One long cable act as a backbone to link all the devices in a network. Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core. As a signal travels along the backbone, some of its energy is transformed into heat. Therefore, it becomes weaker and weaker as it travels farther and farther. For this reason, there is a limit on the number of taps a bus can support and on the distance between those taps.



**Bus Topology Network**

### Advantages

- Ease of installation
- Bus uses less cabling than mesh or star topologies
- Failure of one computer (node) doesn't affect other nodes.

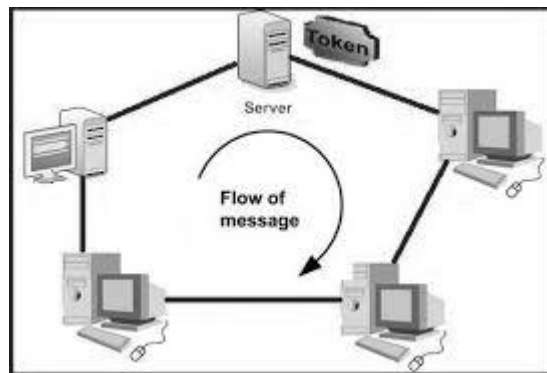
### Disadvantages

- Difficult reconnection and fault isolation
- Signal reflection at the taps can cause degradation in quality.

- Adding new devices may therefore require modification or replacement of the backbone.
- Fault or break in the bus cable stops all transmission, even between devices on the same side of the problem. The damaged area reflects signals back in the direction of origin, creating noise in both directions.

## Ring Topology

In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.



## Advantages

- Easy to install and reconfigure
- To add or delete a device requires changing only two connections.
- Fault isolation is simplified

## Disadvantages

- Unidirectional data transfer
- Break in the ring (such as a disabled station) can disable the entire network

## Hybrid Topology

A hybrid topology is a type of network topology that uses two or more differing network topologies. These topologies can include a mix of bus topology, mesh topology, ring topology, star topology, and tree topology. Hybrid topology inherits merits and demerits of all the incorporating topologies.

Most WANs are connected by means of Dual-Ring topology and networks connected to them are mostly Star topology networks. Internet is the best example of largest Hybrid topology.

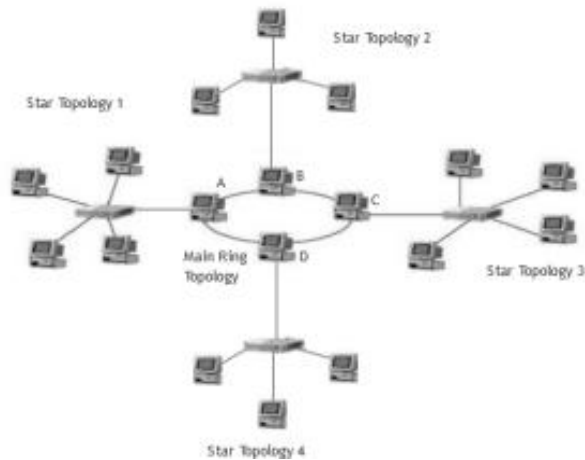


Figure: Hybrid Topology

### Advantages

- Combines the benefits of different types of topologies in one topology.
- Can be modified as per requirement.
- It is extremely flexible.
- It is very reliable.
- Error detecting and troubleshooting is easy.
- Handles large volume of traffic.
- It is used for create large network.

### Disadvantages

- It is expensive.
- Design of a hybrid network is very complex.
- Usually hybrid architectures are usually larger in scales so they require a lot of cables in installation process.
- Installation is a difficult process.

## Basic Concept OSI Reference Model

International Standards Organization (ISO), established in 1947, is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s. An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of

the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.

The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

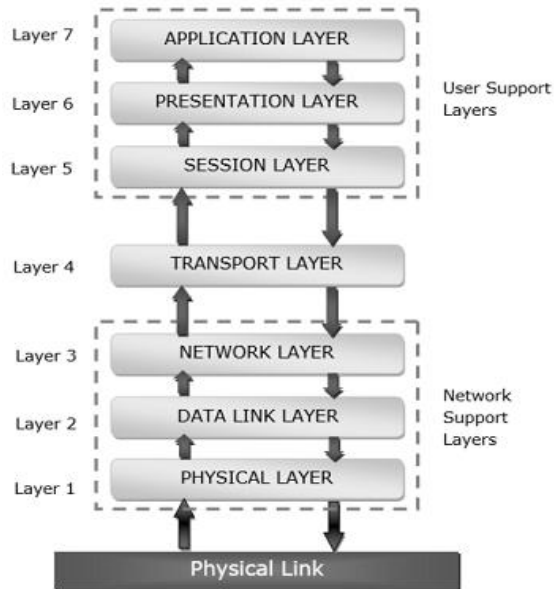
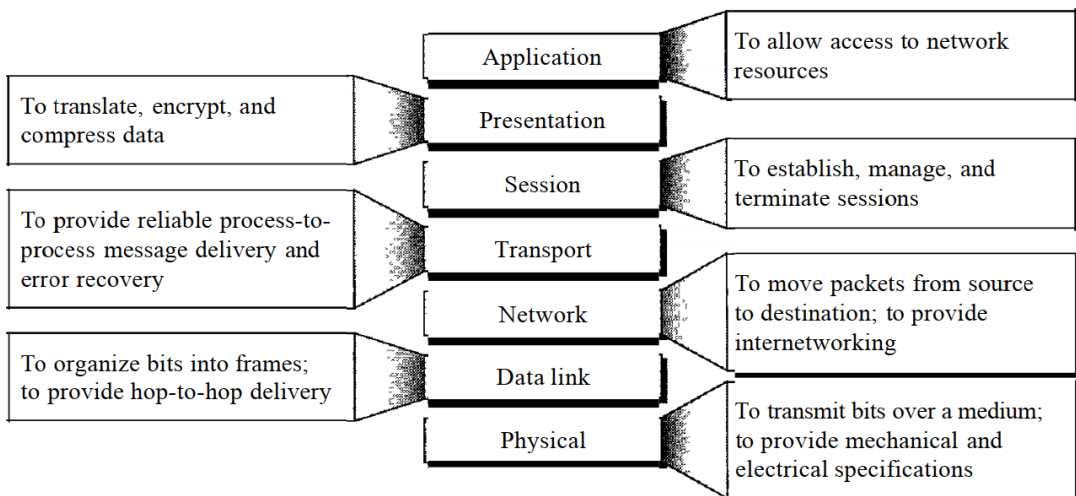


Figure: OSI Reference Model

- Physical Layer:** The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur. The physical layer defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex.
- Data Link Layer:** The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). The data link layer divides the stream of bits received from the network layer into manageable data units called frames. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames.

3. **Network Layer:** The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.
4. **Transport Layer:** The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.
5. **Session Layer:** The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network dialog controller and responsible for synchronization. It establishes, maintains, and synchronizes the interaction among communicating systems.
6. **Presentation Layer:** The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. The presentation layer is responsible for translation, compression, and encryption.
7. **Application Layer:** The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services. The application layer is responsible for providing services to the user.



## Internet Protocol Addressing

A computer somewhere in the world needs to communicate with another computer somewhere else in the world. Usually, computers communicate through the Internet. The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer. For this level of communication, we need a global addressing scheme; IP address in the network layer of the TCP/IP protocol suite.

The Internet addresses are 32 bits in length; this gives us a maximum of  $2^{32}$  addresses. These addresses are referred to as IPv4 (IP version 4) addresses. The need for more addresses, in addition to other concerns about the IP layer, motivated a new design of the IP layer called IPv6 (IP version 6). In this version, the Internet uses 128-bit addresses that give much greater flexibility in address allocation.

### IPv4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet. IPv4 addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time. The IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet. IPv4 uses 32-bit addresses, which means that the address space is  $2^{32}$  or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet. There are two prevalent notations to show an IPv4 address:

- **Binary Notation**

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So, it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. Example: 1110101 10010101 00011101 00000010

- **Dotted decimal notation**

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. Example: 192.168.24.7

### IPv6 ADDRESSES

An IPv6 address consists of 16 bytes (octets); it is 128 bits long. To make addresses more readable, IPv6 specifies hexadecimal colon notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon. To make addresses more readable, IPv6 specifies hexadecimal colon notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.

Example, FDEC: 0074:0000:0000:0000:BOFF:0000:FFFF

### Point to Remember

- Attenuation is the reduction in the strength of analog or digital signal as it is transmitted over a communication medium.
- Distortion means that the signal changes its form or shape.
- Noise is an electrical signal which interferes with an information signal. Also, noise is an error or undesired random disturbance of a useful information signal.
- Amplification is the process to strengthen the amplitude of the signals using an electronic circuit.
- The repeater extends the range of communication systems by re-generating the signals, allowing it to travel further.
- Modulation is the phenomenon of superimposing a message signal with a carrier wave.
- Modulations are of: Amplitude Modulation (AM), Frequency Modulation (FM) and Phase Modulation (PM).
- Amplitude modulation is a technique used in electronic communication, most commonly for transmitting information via a radio carrier wave. Amplitude Modulation works by varying the strength of the transmitted signal in relation to the information being sent.
- Frequency modulation conveys information over a carrier wave by changing the cycles of frequency. The amplitude and phase of the carrier signal are not changed at all. This method has less amount of noise and gives the best encoding method as far as quality is concerned.
- In phase modulation, the amplitude and frequency of the carrier signal remains unchanged but alters its phases. Phase modulation is a form of modulation that can be used for radio signals used for a variety of radio communications applications.
- Demodulation is the reverse process of modulation. It takes a modulated signal and then extracts the original message from it.
- There are five elements of Data Communication/Transmission are: Message, Sender, Receiver, Transmission Medium and Protocol.
- The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
- The sender is the device that sends the data message.
- The receiver is the device that receives the message
- The transmission medium is the physical path by which a message travels from sender to receiver.
- A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating.

- Communication mode (Transmission Mode) refers to the mechanism of transferring of data between two devices connected over a network.
- There are three types of communication mode: Simplex mode, Half-Duplex and Full-Duplex mode.
- In simplex mode, the communication is unidirectional, as on a one-way street.
- In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa.
- In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously. The full-duplex mode is used when communication in both directions is required all the time.
- The network categories are: Local- Area Network (LAN), Wide-Area Network (WAN) and Metropolitan Area Networks (MAN).
- A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus.
- A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world.
- A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city.
- Transmission medium is a path through which information can be transmit from a sender to receiver. There are two main categories of transmission medium: Guided and Unguided Transmission Medium.
- Guided transmission media are also called bounded media or wired media. Guided medium provide conduit from one device to another, the signal is directed and contained by the physical limits of the medium.
- Unguided medium is also referred to as Wireless or Unbounded transmission media. No physical medium is required for the transmission of electromagnetic signals. Uses the section of electromagnetic spectrum known as radio communication (3 KHz - 300 GHz), which is divided into eight bands, to transmit signals.
- Jitter is a problem if different packets of data encounter different delays and the application using the data at the receiver site is time-sensitive such as audio and video data.
- Singing occurs as a result of transmitted signals being coupled into a return path and fed back to the respective sources.
- Echo occurs when some portion of the signal is returned/arrived with enough delay (typically more than about 30 ms) to make the signal distinguishable from normal sidetone.

- Crosstalk is the effect of one wire on the other. One wire acts as a sending antenna and the other as the receiving antenna. Crosstalk may affect both wires and create unwanted signals.
- Bandwidth is the rate at which digital data can be transmitted over a given communication channel.
- Network architecture is the logical and structural layout of the network, consisting of transmission equipment, software and communication protocols, and infrastructure (i.e. wired or wireless) transmission of data and connectivity between components.
- There are two types of network architecture is: Client Server and Peer-to-peer.
- Client-server architecture is a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer).
- In a peer-to-peer network, tasks are allocated to every device on the network. Furthermore, there is no real hierarchy in this network, all computers are considered equal and all have the same abilities to use the resources available on this network.
- An IP address is a unique address that identifies a device on the internet or a local network.
- Subnet is a logical subdivision of an IP network. The practice of dividing a network into two or more networks is called subnetting. Subnetting increases routing efficiency.
- A MAC address, or Media Access Control address, is a 48- or 64-bit address associated with a network adapter. While IP addresses are associated with software, MAC addresses are linked to the hardware of network adapters.
- An internet is the global system of interconnected computer networks to communicate between networks and devices.
- An intranet is a private network (LAN) that uses the Internet model. However, access to the network is limited to the users inside the organization.
- An extranet is the same as an intranet with one major difference: Some resources may be accessed by specific groups of users outside the organization under the control of the network administrator.
- The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. There are four basic topologies: Mesh, Star, Bus and Ring topology.
- In a mesh topology, every device has a dedicated point-to-point link to every other device.
- In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to one another.
- A bus topology, on the other hand, is multipoint. One long cable act as a backbone to link all the devices in a network.

- In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it.
- A hybrid topology is a type of network topology that uses two or more differing network topologies. These topologies can include a mix of bus topology, mesh topology, ring topology, star topology, and tree topology.
- The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems.
- There are seven layers in OSI model, they are: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer and Application Layer.
- The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium.
- The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer).
- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links).
- The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host.
- The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network dialog controller and responsible for synchronization.
- The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.
- The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

### ✓ Multiple Choice Questions

1. A computer network permits sharing of ....
  - a. Resources
  - b. Information
  - c. Both A and B
  - d. None of these
2. \_\_\_\_\_ is the technology that connects the machines and people within a site in a small area.
  - a. LAN
  - b. MAN
  - c. WAN
  - d. None of these
3. \_\_\_\_\_ is a network that covers geographic areas that are larger, such as districts or cities.
  - a. LAN
  - b. MAN
  - c. WAN
  - d. None of these
4. \_\_\_\_\_ is a network that connect sites that are in diverse locations.
  - a. LAN
  - b. MAN
  - c. WAN
  - d. None of these
5. \_\_\_\_\_ is a collection of point to point links that may form a circle.
  - a. Bus Topology
  - b. Ring Topology
  - c. Mesh Topology
  - d. Star Topology

6. Which protocol layer uses the protocols are WWW, HTTP, FTP, SMTP, e-mail, etc.
- Application Layer Protocol
  - Transport Layer Protocol
  - Internet Layer Protocol
  - None of these
7. Devices may be arranged in a \_\_\_\_\_ topology.
- Mesh
  - Ring
  - Bus
  - All of the above
8. \_\_\_\_\_ is a collection of many separate networks.
- A WAN
  - An Internet
  - A LAN
  - None of these
9. A \_\_\_\_\_ is a set of rules that governs data communication.
- forum
  - standard
  - portocol
  - None of these
10. The \_\_\_\_\_ is the physical path over which a message travels.
- protocol
  - medium
  - signal
  - None of these
11. The information to be communicated in a data communications system is the \_\_\_\_\_.
- medium
  - protocol
  - message
  - transmission
12. Frequency of failure and network recovery time after a failure are measures of the \_\_\_\_\_ of a network.
- performance
  - reliability
  - security
  - feasibility
13. An unauthorized user is a network \_\_\_\_\_ issue.
- performance
  - reliability
  - security
  - All of the above
14. Communication between a computer and a keyboard involves \_\_\_\_\_ transmission.
- simplex
  - half-duplex
  - full-duplex
  - automatic
15. A television broadcast is an example of \_\_\_\_\_ transmission.
- Half-duplex
  - Full-Duplex
  - Simplex
  - None of these
16. In a \_\_\_\_\_ connection, more than two devices can share a single link.
- point-to-point
  - multipoint
  - primary
  - secondary
17. Why was the OSI model developed?
- Manufacturers disliked the TCP/IP protocol suite
  - The rate of data transfer was increasing exponentially
  - Standards were needed to allow any two systems to communicate
  - None of the above
18. The \_\_\_\_\_ model shows how the network functions of a computer ought to be organized.
- CCITT
  - ISO
  - OSI
  - ANSI
19. The OSI model consists of \_\_\_\_\_ layers.
- five
  - six
  - seven
  - eight
20. In OSI model, what is the main function of the transport layer?
- node-to-node delivery
  - process-to-process message delivery
  - synchronization
  - updating and maintenance of routing tables
21. In the OSI model, encryption and decryption are functions of the \_\_\_\_\_ layer.
- transport
  - session
  - presentation
  - application
22. When a host on network A sends a message to a host on network B, which address does the router look at?
- Port
  - Logical
  - Physical
  - None of these
23. IPv6 has \_\_\_\_\_ -bit address.
- 32
  - 64
  - 128
  - variable
24. The \_\_\_\_\_ layer is responsible for moving frames from node to the next.
- physical
  - data link
  - transport
  - None of these
25. The \_\_\_\_\_ layer is responsible for the delivery of a message from one process to another.
- physical
  - data link
  - transport
  - None of these
26. The \_\_\_\_\_ address, also known as the link address, is the address of a node as defined by its LAN or WAN.
- port
  - physical
  - logical
  - None of these

27. The seven-layer \_\_\_\_ model provides guidelines for the development of universally compatible networking protocols.
- a. OSI                      b. ISO  
c. IEEE                     d. None of these
28. The physical, data link and network layers are the \_\_\_\_ support layers.
- a. user                      b. network  
c. both (a) and (b)      d. neither (a) nor (b)
29. The session, presentation and application layers are the \_\_\_\_ support layers.
- a. user                      b. network  
c. both (a) and (b)      d. neither (a) nor (b)
30. The \_\_\_\_ layer coordinates the functions required to transmit a bit stream over a physical medium.
- a. transport                b. network  
c. data link                d. physical
31. The \_\_\_\_ layer is responsible for delivering data units from one station to the next without errors.
- a. transport                b. network  
c. data link                d. physical
32. The \_\_\_\_ layer is responsible for the source-to-destination delivery of a packet across multiple network links.
- a. transport                b. network  
c. data link                d. physical
33. The \_\_\_\_ layer establishes, maintains, and synchronizes the interactions between communicating devices
- a. transport                b. session  
c. data link                d. physical
34. The \_\_\_\_ layer ensures interoperability between communicating devices through transformation of data into a mutually agreed upon format.
- a. transport                b. session  
c. data link                d. presentation
35. The \_\_\_\_ layer enables the users to access the network
- a. transport                b. application  
c. data link                d. physical

**Answer Key**

1.c	2.a	3.b	4.c	5.b	6.a	7.d	8.b	9.c	10.b
11.c	12.b	13.c	14.a	15.c	16.b	17.c	18.c	19.c	20.b
21.c	22.b	23.c	24.b	25.c	26.b	27.a	28.b	29.a	30.d
31.c	32.b	33.b	34.d	35.b					

 **Exercise**

■ **Short Answer Questions**

1. What is transmission media? Explain any two different types of transmission media.
2. What is optical fiber? Write advantages of optical fiber cable.
3. Differentiate between LAN and WAN.
4. What is communication system? Explain basic elements of communication system.
5. Explain Unshielded Twisted Pair (UTP) and Coaxial cable.
6. Describe star topology with suitable diagram.
7. Explain bus and ring topology with advantages and disadvantages.
8. Differentiate between guided and unguided networking media.
9. Differentiate between internet and intranet.
10. Describe different types of communication modes.
11. Write the difference between Client server and peer to peer Network
12. Differentiate between IPV4 and IPV6

13. Write short notes on:
  - g. Transmission Medium
  - h. OSI Model
  - i. MODEM
  - j. Optical Fiber
  - k. IP Address

### ■ Long Answer Questions

1. What are topologies? Explain different types of topology.
2. What is OSI Model? Explain.
3. Explain the different network connecting devices.

### Project Work

1. Install a Packet Tracer software and perform the following activities:
  - a. Create a LAN using star topology. Add at least 5 computers on a network
  - b. Also perform an initial configuration if possible.
  - c. Create a WAN Connections and configure if possible
2. Turn on remote desktop on your computer by opening Setting-> System-> Remote Desktop. Use Remote Desktop Connection to remotely access and control another PC.





## Web Technology II



### LEARNING OBJECTIVES

After going through this unit, you will be able :

- ◆ know about the latest web development technologies
- ◆ know about the languages used in modern web pages
- ◆ developing dynamic web pages with the use of JavaScript and PHP
- ◆ know about the different scripting techniques
- ◆ develop the basic programming skills using JavaScript and PHP
- ◆ learn event handling techniques
- ◆ validate the form data using JavaScript
- ◆ learn about server side scripting- PHP
- ◆ establish database connection using server side scripting
- ◆ display data in tables from database

## Introduction

Web is a networked information system that contains a huge collection of files. The contents of the files are written in HTML or other programming languages. In class XI you study how to develop simple pages using HTML. You have created static web pages using different HTML tags. You already know how to create links, create table, insert image, audio, video, and create forms. To add styles you have used CSS in the HTML code. This knowledge of HTML and CSS can only design static web pages. To design dynamic web pages you must have knowledge of server side programming and database concept. Now in this topic you will learn about server side and client side scripting languages.

Web technology works on client server system where client is a browser and server is a web server. Client requests the web pages to the server and server respond to those requests. HTML and other files are sent from servers to client. Many numbers of clients can request for the same resource. Web server responds to all those clients equally.

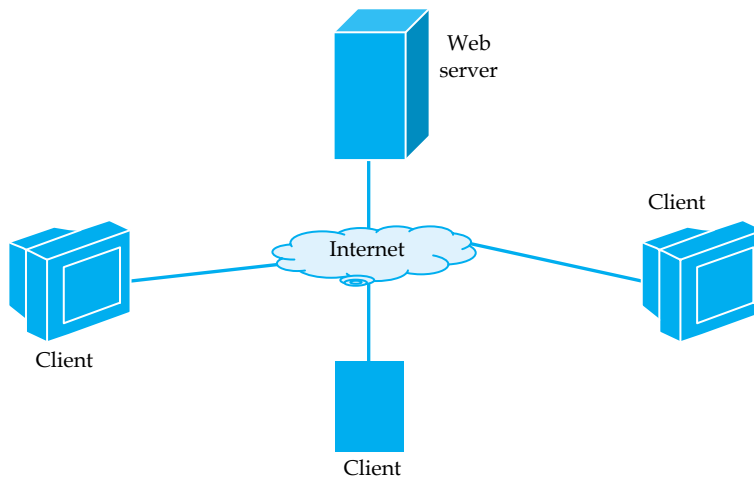


Figure: Client server web technology

## Server Side and Client Side Scripting

Scripting language (also known as scripting, or script) is a series of commands that are executed without being compiled. Scripting languages are a specific kind of computer languages that you can use to give instructions to other software, such as a web browser, server, or standalone application. Many of today's most popular coding languages are scripting languages, such as JavaScript, PHP, Ruby, Python, and several others.

A scripting language is a programming language designed for integrating and communicating with other programming languages. Scripting language is used to create dynamic web pages. The scripting language is basically a language where instructions are written for a run time

environment. They do not require the compilation step and are rather interpreted. The server side and client side scripting concept is illustrated in the figure below:

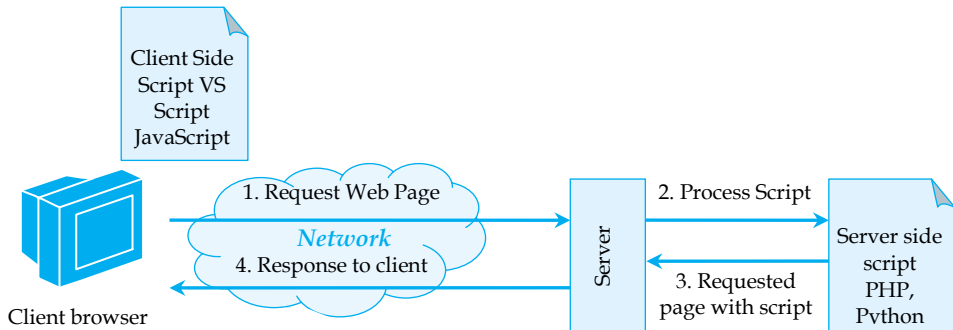


Figure: Client side and server side scripting

### Application Areas of Scripting Language

- Dynamic Web Applications
- Game Application and Multimedia
- Scripting like R and Python are used in statistics and research
- To automate the process
- Used to create plug ins and extensions for existing applications

Some of the Popular Languages are

1. **Python:** Python is a very popular and demanding programming language now because it is suitable for developing very simple to complex applications. It is also used to connect database systems. Python has simple English like syntax.
2. **Perl:** Practical Extraction and Reporting Language, first released in 1987 is a powerful language with advance features. Perl is a stable, cross platform programming language. It is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.
3. **Ruby:** Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.
4. **Bash:** A Bash script is a plain text file which contains a series of commands. It is widely available on various operating systems and is a default command interpreter on most GNU/Linux systems.
5. **Node.js:** Node.js is used to create dynamic page contents. It can create, open, read, write, delete, and close file on the server.
6. **ASP.net:** It is used to develop dynamic websites, web applications, and web services. ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites.

7. **VBScript:** Visual Basic Script (VBScript) is an open source web programming language developed by Microsoft. It is superset of JavaScript and adds optional static typing class-based object oriented programming. VBScript is lightweight scripting language. As VBScript is only used by IE browsers JavaScript is preferred over VBScript.
8. **JavaScript:** JavaScript is most well known and widely used scripting language for web pages. All java script files are stored in file having .js extension. JavaScript and Java programming language are two different things. JavaScript is generally used for making websites interactive and dynamic.
9. **jQuery:** JQuery is a JavaScript library that simplifies writing code and enables rapid web development. jQuery simplifies HTML document traversing and manipulation, and browser event handling. The main concept of jQuery is "write less, do more".
10. **PHP:** Hypertext Preprocessor (PHP) is widely used scripting language. PHP scripts are executed on the server. It is used to manage dynamic content, databases, session tracking, and building e-commerce sites. It is integrated with a popular database MySQL.

### Advantages of Scripts

- Easy to learn and write
- A scripting language is a lightweight programming language
- Much faster to develop the program
- Scripting language makes coding simpler and faster
- Scripting languages are widely used in web development.
- Editing of code is faster
- It can help to develop enhanced web pages

### Types of Scripting Languages

Scripting language can be divided into two categories:

- Server Side Scripting Languages
  - Client Side Scripting Languages
1. **Server Side Scripting Languages:** Server-side scripting languages create the scripts that run on the server and hence minimize the workload of a client browser. The most commonly used server-side scripting languages are Perl, Ruby, Python, PHP, JSP, etc. The each and every script is interpreted by an application on the server. The application which interprets the server side scripts is installed on the server just like any other application. Server side scripts can communicate with database using a structures query language (SQL) which performs different operations (such as: insertion, deletion, modification, etc) on database.

#### Advantages of Server side scripting:

- You can create dynamic pages
- Can connect to databases that resides on the web server

- Can access files from the server to client browser
- Users are not able to block the contents from server
- The actual code is not visible to the client
- Authentication and verification of user is possible
- It supports many databases like: MySQL, Oracle,
- Efficient storage and delivery of information
- Customized user experience
- Controlled access to content
- Notification and communication
- Users do not need to download plug-in like java or flash
- The content management system (CMS) makes editing simpler.

#### Disadvantages of Server side Scripting:

- The scripting software has to be installed on the server
- The script takes more time to execute.
- It requires a large amount of memory space in the server computer.
- Implementation cost is high
- If a lot of users are accessing server data, server may crash due to overload

2. **Client Side Scripting Languages:** Client-side scripting languages create the scripts that run on the client side (browser). Some good examples are JavaScript, jQuery etc. Client-side scripting refers to the programs that are executed on client-side. Client-side scripts contain the instruction for the browser to be executed in response to certain user's action. To develop an interactive and eye catching web site HTML only is not sufficient. HTML is good for developing static web pages that includes image, text, hyperlinks, presentations, etc. Such static web page looks the same each time we visit the site. In order to develop the interactive and dynamic web pages you must integrate some scripting languages. Client side scripting languages are embedded in HTML of a web page to create the dynamic output so that user views different contents and information when he visits the page.

#### Advantages of Client side Scripting:

- Immediate response to users
- Enhance the appearance of websites
- More responsive design and interaction with the user
- It does not need to send requests to the server hence reduces the load on server
- Loading time of a page is faster
- Reduces the network traffic
- It is reusable

#### Disadvantages of Client side Scripting:

- All browsers may not support client side script
- The code is not secure because anyone can look at the code

- Users can disable the client side scripts so required content may not be displayed
- Database connection is not possible with client side scripting.
- Dynamic content cannot be displayed.

### Difference between Scripting Language and Programming Language

Basis of Comparison	Scripting Language	Programming Language
platform used	Scripting languages are platform-specific	Programming languages are platform-independent
Compilation	Most of the scripting languages are interpreted	Most of the programming language are compiled
Speed	Scripting languages runs slower than programming languages.	Programming languages runs faster than scripting language
Length of code	Developer has to write less code compared to programming language	Developer has to write much code compared to scripting language
Development of Application	We cannot create standalone application with scripting language only.	We can create standalone application with programming language only.
Examples	JavaScript, VBScript, Python, Perl, ASP etc.	C, C++, Java, etc.
Script Running	Scripting languages run inside other programs. It is dependent on other programming language	It is not dependent on other programs to run. It is independent.

### Difference between Client Side and Server Side Scripting Language

Basis of Comparison	Server side Scripting	Client side Scripting
Execution of code	The server executes the server side scripting	The client(web browser) executes the client side scripting
Database Processing	Can be used to connect database on the web server	Cannot be used to connect to the database on the web server
Speed	Server side scripting response is slower	Client side scripting response is faster
Security	Source code is not visible to the user so it is secure	Source code is visible to user so it is relatively insecure
Users control	Users can not block server side scripting	Users can block client side scripting
Examples	Examples: PHP, ASP.NET, ASP , Ruby on rails, Python, Perl	Examples: JavaScript, VBScript
Dependency	Does not depend on client. Any server side technology can be used	Depends on browser and version of the browser

## Introduction of Internet Technology

Internet technology is the technology that allows users to access information over the World Wide Web (WWW). Internet and web technology is changing so fast that you can view the contents on the web with elegant graphic design and in interactive way. There are number of websites to visit for various contents on the internet. The websites are commonly categorized into two types. They are:

### Static Websites

Websites that contain stable contents that are displayed using web pages are called static websites. In static website all the pages provide the same information to all the visitors. Static web site is a simple type of site where content are not changed according to user's interaction. Static websites are more secure because there is no need of database connectivity. Static websites are also loads faster and are easy to design. Generally static websites are developed to display same information to every user. But nowadays dynamic websites are more popular because the contents on the site are changing time to time so user can view different contents on the web page. Static websites is fine for small and fixed contents which do not change over time. A static web page is ready before it is accessed. Some uses of static websites are:

- Documentation
- Maintenance page
- Landing page
- Blogs
- Web site Presentation

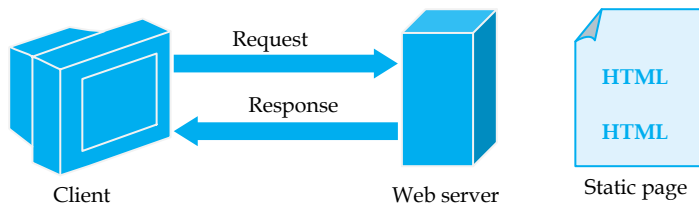


Figure: Static Web site

### Dynamic Websites

Dynamic website is a collection of dynamic web pages whose content changes dynamically. It accesses content from a database or Content Management System (CMS). Therefore, when you alter or update the content of the database, the content of the website is also altered or updated. It uses client side scripting or server side scripting or both to generate dynamic contents. Dynamic websites are used where the information is used frequently. For example: stock prices, whether information, news etc. The examples of dynamic websites are: Facebook, YouTube, news portals, etc. The content of dynamic web page is generated each time it is accessed.

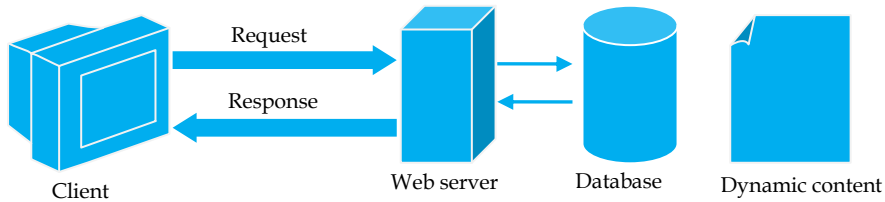


Figure: Dynamic Web site

## JavaScript Fundamental

JavaScript is a dynamic scripting language (website development environment). It is lightweight and most commonly used as a part of web pages. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is used to make web pages interactive. It runs on your visitor's computer and doesn't require constant downloads from your website.

### Advantages of JavaScript

The advantages of using JavaScript are

1. **Less server interaction:** You can validate user input before sending the page off to the server. This saves server traffic, which means fewer loads on your server.
2. **Easy to learn:** By learning few commands and simple rules of syntax, you can easily build applications using JavaScript.
3. **Immediate feedback to the visitors:** They don't have to wait for a page reload to see if they have forgotten to enter something.
4. **Increased interactivity:** You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
5. **Quick Development:** Scripts can be developed in short period of time
6. **Richer interfaces:** You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.
7. **Transmitting information:** About the user's reading habits and browsing activities to various websites. Web pages frequently do this for web analytics, ad tracking, personalization or other purposes.
8. **Easy Debugging and Testing-** As JavaScript is interpreted line by line, it is easy to find error and make changes.
9. **Interactive content,** for example games, and playing audio and video
10. **Validating input values** of a Web form to make sure that they are acceptable before being submitted to the server.

## Adding JavaScript to HTML Page

JavaScript itself cannot do anything so it should be embedded into a standard HTML program. JavaScript is embedded between `<SCRIPT> ..... </SCRIPT>` HTML tags. You can place the `<SCRIPT>` tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the `<HEAD>` tags.

A simple syntax of your JavaScript will appear as follows.

```
<script >
  JavaScript code
</script>
```

The script tag takes two important attributes –

Attribute	Meaning
Language	This attribute specifies what scripting language you are using. Typically, its value will be JavaScript.
Type	This attribute is recommended to indicate the scripting language in use and its value should be set to "text/JavaScript".

**For example:**

```
<html>
  <body>
<script language = "JavaScript" type = "text/JavaScript">
  document.write("Hello World!")
  </script>
</body>
</html>
```

\* Semicolons are Optional in JavaScript; however it is a good programming practice to use semicolons.

\* JavaScript is a case-sensitive language.

## JavaScript Data Types

Data types specify what kind of data can be stored and manipulated within a program. A data type in a language defines the type of data a variable can hold. Although there are different data types in JavaScript, you can put any value to a variable. For example,

```
let n=7; //decimal number
n=15.5; //floating point number
let age="twenty five"; //string
var colors = ["Red","Green","Blue"]; //Array
```

In the above example, the value of variable `n` represents both integer and float. Similarly, the value of variable `age` represents both string and integer. It is absolutely allowed in JavaScript. This is why JavaScript is also called "dynamically typed" language. It means there are different data types exist but variables are not bound to any of them.

Data Types of JavaScript are of following types:

## 1. Primitive Data types

- Number
- String
- Boolean
- NULL

## 2. Non primitive

- Objects
- Array
- Function

### Number:

A number data type can be an integer, a floating point value, an exponential value, a NaN(Not a Number) or a 'Infinity'.

- Integer numbers can be represented in decimal, hexadecimal and octal form.
- Floating point numbers are used to represent numbers that requires the use of decimal points.
- Floating point numbers can represent exponential notations also.
- For example: 22,11.8, 3E8, 0xAF

```
var x=20; // integer value
var y=15.5; // a number containing a decimal
var z= 10e2; // an exponential value
```

### String:

The string data type in JavaScript can be any group of characters enclosed in a single or double-quotes. A string is a sequence of one or more characters that are enclosed by double (") or single (') quotes. If a string begins with a single quote it must end with single quote. If a string begins with a double quote it must end with double quote. For example:

```
var name = "Devendra Chapagain"; //using double quote
var university="Lovely Professional University";//using double quote
var qualification= 'Master in Computer Application'; // using single quote
```

### Boolean:

The Boolean data type is mostly used to check a logical condition. It has only two values, TRUE and FALSE. Thus Booleans are logical data types which can be used for comparison of two variables or to check a condition. JavaScript automatically converts the Boolean values TRUE and FALSE into 1 and 0 when they are used in numerical expression. For example:

```
var a =3;
var b=6;
if(a<b)
{
  alert(a is a smaller number than b);
}
```

### NULL:

A NULL value means no value, which identifies an empty. The null in JavaScript is a data type that is represented by only one value, the 'NULL' itself. Use of NULL value prevents the sort of errors that result from using un-initialized variables. It's just a special value which represents "nothing", "empty" or "value unknown". In the example below it states that the value of variable 'a' is unknown.

```
var a = null;
```

### Array:

An array can hold more than one value at a time. Arrays are used to store multiple values in a single variable. An array in JavaScript is an object data type. An array contains more than one value with an integer index, where the index starts from 0. JavaScript arrays are written with square brackets. Array items are separated by commas.

#### Syntax to create array:

```
var array_name = [item1, item2, ...];  
or  
var array_name= new array(item1, item2,...);
```

#### For example:

```
<html>  
<body>  
  Displaying Array values using for loop  
<script>  
  var a = ["a", "b", "c"];  
  for(i=0;i<3;i++)  
  {  
    document.write(a[i]);  
  }  
</script>  
</body>  
</html>
```

### Function:

A JavaScript function is a block of code designed to perform a particular task. A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions.

#### Syntax to create a function:

```
<script type = "text/javascript">  
function function_name(parameter-list)  
{  
  statements;  
}  
</script>
```

We will discuss more about function later.

## Variables and Operators

### JavaScript Variables

Variables are the names you give to computer memory locations which are used to store values in a computer program. The values of variable can be changed during the program execution. Before you use a variable in a JavaScript program, you must declare it. JavaScript is un-typed language. This means that a JavaScript variable can hold a value of any data type. Unlike many other languages, you don't have to tell JavaScript during variable declaration what type of value the variable will hold. The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.

Rules for JavaScript variable names:

- Variable names are case sensitive (a and A are two different variables)
- Variable names must begin with a letter or the underscore character
- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names cannot start with numbers
- Names can also begin with \$ and \_
- Reserved words cannot be used as names
- JavaScript variables are declared using var or let keyword
- Variables can have short names (like a and b) or more descriptive names (like total\_average, age)
- Variable name cannot contain spaces.

JavaScript variables have only two scopes.

- **Global Variables:** A global variable has global scope which means it can be defined anywhere in your JavaScript code.
- **Local Variables:** A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

Variables are declared to store data by using two keywords: *var* and *let*. You can assign a value to a variable using equal to (=) operator during declaration.

Var - is an old-school variable declaration.

Let- is a modern variable declaration.

**For example:**

```
<script type="text/javascript">
var x= 5; // Number
let name= "Aayan Chapagain"; // String
</script>
```

## JavaScript Operators

Operators are symbols that tell the compiler to perform some specific tasks. For example,  $a+b$  where '+' sign is an operator and 'a' and 'b' are operands and '+' sign indicates addition operation between the operands.

JavaScript supports the following types of operator:

- Arithmetic Operators
- Relational Operators
- Logical Operator
- Assignment Operators
- Conditional (or ternary) Operators
- Increment/Decrement Operators
- Bitwise Operators

### Arithmetic Operators

The operators that are used to perform mathematical calculation like addition, subtraction, multiplication and division are called arithmetic operators. Arithmetic operator is a binary operator that operates on two operands.

Operator	Operation	Description	Example	Result (let a=5, b=3)
+	Addition	Adds two operands	a+b	8
-	Subtraction	Subtract second operand from first operand	a-b	2
*	Multiplication	Multiply two operands	a*b	15
/	Division	Divide first operand by second	a/b	1.6
%	Modulus	Gives remainder after division	a%b	2

#### Example:

```
<html>
<body>
  <h1>Arithmetic operator Example</h1>
  <script type="text/javascript">
    var a = 22;
    let b = 5;

    sum = a+b; //arithmetic Addition
    dif=a-b; //arithmetic Subtraction
    document.write("Sum="+sum+"<br>");
    document.write("Difference="+dif);

  </script>
</body>
</html>
```

#### Result:

#### Arithmetic Operator Example

Sum = 27

Difference = 17

## Relational Operators

The Relational operators are used for comparison. The result of relational operator is always either TRUE (1) or FALSE (0) based on the comparison result. The relational operators are as follows:

Operator	Operation	Description	Example	Result (let x=1, y=2)
==	Equal to	Checks if the values of two operands are equal or not. If both values are equal then it returns TRUE	x==y	0
!=	Not Equal to	Checks if the values of two operands are equal or not. If both values are not equal then it returns TRUE	x!=y	1
<	Less than	Checks if value of first operand is smaller than second.	x<y	1
<=	Less than or equal to	Checks if value of first operand is smaller than or equal to second.	x<=y	1
>	Greater than	Checks if values of first operand is greater than second.	x>y	0
>=	Greater than or equal to	Checks if a value of first operand is greater than or equal to second.	x>=y	0

### Example:

```
<html>
<body>
  <h2>Relational operator Example</h2>
  <script>
    var a=5;
    var b=6;
    if(a>b)
      document.write("a is greater than b");
    else
      document.write("b is greater than a");
  </script>
</body>
</html>
```

### Result:

#### Relations Operator Example

b is greater than a

## Logical Operator

Logical operators are used to evaluate expressions which may be TRUE or FALSE. Expressions which involve logical operations are evaluated and found to be one of two state 0 or 1. If you want to test multiple conditions in the process of making a decision, you have to perform simple tests in separate if statements. Logical operators are used to form more complex conditions by combining simple conditions.

Operator	Operation	Description	Example	Result (let a=5, b=3)
&&	AND	Returns TRUE if both expression are TRUE. It is equivalent to AND operator	(a>b) &&(a<8)	0
	OR	Returns TRUE if any one of the expression is TRUE. It is equivalent to OR operator	(a>b) &&(a<7)	1
!	NOT	It is equivalent to NOT operator	a!=b	1

**Example:**

```
<html>
<body>
  <h2>Logical opeator Example</h2>
<script>
  var a=5;
  var b=6;
  c=(a>b) &&(a<6);
  d=(a>b) || (a<6);
  document.write("c="+c+"<br>");
  document.write("d="+d);
</script>
</body>
</html>
```

**Result:****Logical Operator Example**

c = false

d = true

**Conditional (or Ternary) Operators**

Conditional operator also called ternary operator because it takes three operands. The result of ternary operator is evaluated according to the condition.

The syntax of the ternary operator is:

```
(Condition)?(statement 1):(statement 2);
```

- Statement 1 is executed if condition is true
- Statement 2 is executed if condition is false

**Example:**

```
<html>
<body>
  <h2>conditional opeator Example</h2>
<script>
  var a=2;
  var b=3;

  large=(a>b)?a:b;

  document.write("The larger number is: "+large);

</script>
</body>
</html>
```

**Result:****Conditional Operator Example**

The larger number is: 3

## Increment/Decrement Operators

**Increment operator (++):** The token '+' is used twice to represent increment operator. The increment operator is used to increase the value of variable by 1. There are two types of increment operators:

### Post increment

If increment operator '++' appears after the variable name, then it is known as post increment operator. For example, a++, b++ etc. In post increment operator, other operations are performed at first and the increment operation takes at last.

### Pre increment

If increment operator '++' appears before the variable name, then it is known as pre increment operator. For example, ++a, ++b etc. In pre increment operator, increment operation takes at first and other operations are performed later.

**Decrement operator (--):** The token '-' is used twice to represent decrement operator. The decrement operator is used to decrease the variable by 1. There are two types of decrement operators:

- **post decrement:** If decrement operator '--' appears after the variable name, then it is known as post decrement operator. For example, a--, b++ etc. In post decrement operator, other operations are performed at first and the decrement operation takes at last.
- **pre decrement:** If decrement operator '--' appears before the variable name, then it is known as pre decrement operator. For example, --a, --b etc. In pre decrement operator, decrement operation takes at first and other operations are performed later.

### Example:

```
<html>
<body>
  <h2>Increment/Decrement operator Example</h2>
  <script>
    var a=5;
    var b=10;
    a++; //increment operator
    b--; //decrement operator

    document.write("a++=: "+a +"<br>");
    document.write("b--=: "+b +"<br>");

  </script>
</body>
</html>
```

### Result:

#### Increment/Decrement Operator Example

```
a++=: 6
b-- =: 9
```

## Bitwise Operators

Any operators which are used to operate at bits level of any value are called bitwise operators. A bitwise operation operates on one or more bit patterns or binary numerals at the level of their individual bits. The operation process performed after converting every value to binary equivalent. The operators include:

Operator	Operation	Description	Example
&	Bitwise AND	Both operands are converted into binary equivalent and each bit and perform AND operation with corresponding bit.	c=a&b
	Bitwise OR	Both operands are converted into binary equivalent and each bit and perform OR operation with corresponding bit.	d=a b
~	Bitwise NOT	Convert the operand into binary equivalent and invert each of the bits. i.e 0 to 1 and 1 to 0	e=~b
^	Bitwise XOR	Both operands are converted into binary equivalent and each bit and perform XOR operation with corresponding bit.	f=a^b
<<	Left shift	Convert the operand into binary equivalent and Perform left shift operation	g=a<<2
>>	Right shift	Convert the operand into binary equivalent and Perform right shift operation	h=a>>2

### Example:

```
<html>
<body>
  <h2>Bitwise operator Example</h2>
  <script>
    var a=2;
    var b=3;

    c=a&b; //bitwise AND
    d=a|b; //bitwise OR
    e=~a; //bitwise NOT
    f=a<<2; //bitwise left shift with 2
    g=a>>2; //bitwise right shift with 2

    document.write("Bitwise a&b =" +c+"<br>");
    document.write("Bitwise a|b =" +d+"<br>");
    document.write("bitwise NOT =" +e+"<br>");
    document.write("bitwise left shift with 2 =" +f+"<br>");
  </script>
</body>
</html>
```

### Result:

#### Bitwise Operator Example

Bitwise a & b = 2

Bitwise a | b = 3

bitwise NOT = -3

bitwise left shift with 2 = 8

## Functions and Control Structures

### JavaScript Function

A JavaScript function is a block of code designed to perform a particular task. A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions.

#### Advantages of Using Function:

- You can't do anything in JavaScript without function
- Functions increase code reusability.
- Function helps to structure the code properly.
- Functions make code less complex
- Functions make code more readable and extendable
- Functions can be called anywhere in the program
- Functions help program developers to write the modular codes
- Function also allows a programmer to divide a big program into a number of small and manageable functions

#### Function Definition

Before we use a function, we need to define it. The most common way to define a function in JavaScript is by using the **function** keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

#### Syntax:

```
<script type = "text/javascript">
function function_name(parameter-list)
{
    //statements
}
</script>
```

#### Calling a Function

The code written inside a function does not execute unless it is called. To call a function somewhere later in the script, you would simply need to write the name of that function as shown in the following code.

```
<html>
<head>
<script type = "text/javascript">
function sayHello()
{
    document.write ("Hello Devendra !");
}
</script>
</head>
<body>
```

```

<p>Click the following button to call the function</p>
<form>
  <input type = "button" onclick = "sayHello()" value = "Say Hello">
</form>
</body>
</html>

```

## Function Parameters

A function can take multiple parameters separated by comma. The function parameter can have value of any data type.

For example:

```

<html>
<body>
  <h2>Function with Parameter</h2>
<script>
  function add(a, b)
  {
    var sum=a+b;
    document.write("Sum="+sum);
  }
  // calling functions
  add(3,4);
</script>
</body>
</html>

```

**Result:**

**Function with Parameter**

Sum = 7

## JavaScript Control Structures

A control structure refers to the flow of execution of the program. These control structures enables us to specify the order in which the various instructions in a program are to be executed. The types of control structure are:

1. Selection Control structure
  - If
  - If else
  - If else if
  - Switch case
2. Looping Control structure
  - For loop
  - While loop
  - Do while loop

**If statement:** The **if** statement is the fundamental control statement that allows JavaScript to make decisions and execute statements conditionally.

**Syntax:**

```
if(condition)
{
Statement(s) to be executed if condition is true
}
```

**Flowchart:**

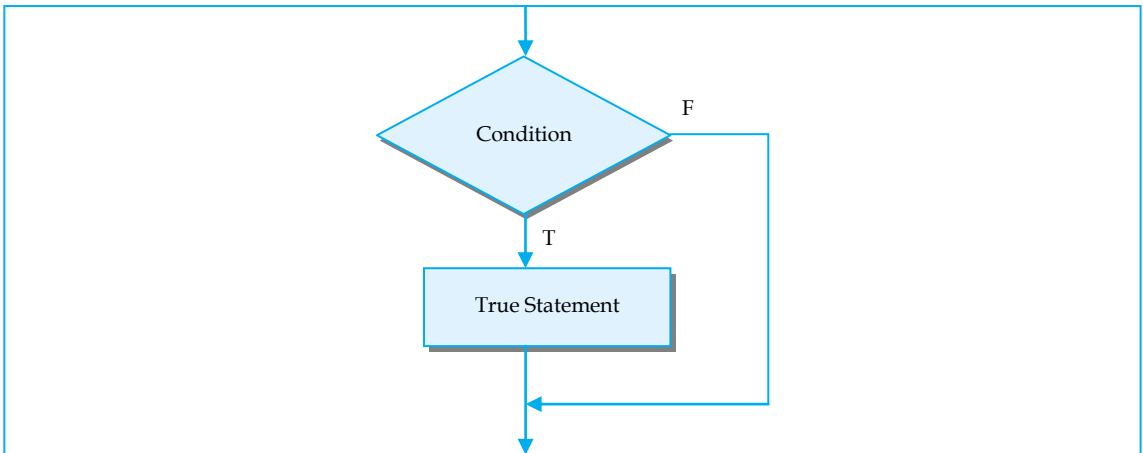


Figure: Flowchart of if statement

**Example:**

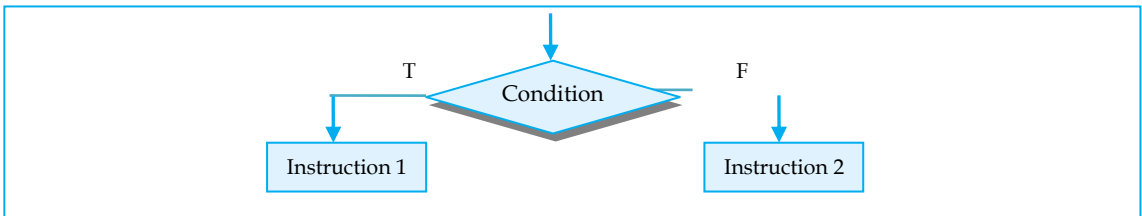
```
<html>
<body>
<script type = "text/javascript">
  var age = 20;
  if(age > 18)
  {
  document.write("<b>Qualifies for driving</b>");
  }
</script>
</body>
</html>
```

**If else**

The '**if...else**' statement is the next form of control statement that allows JavaScript to execute statements in a more controlled way.

**Syntax :**

```
if(expression)
{
Statement(s) to be executed if expression is true
}
else
{
Statement(s) to be executed if expression is false
}
```

**Flowchart:****Example:**

```

<html>
<body>
<script type = "text/javascript">
  var x = 5,y=8;
  if( x>y )
  {
    document.write("<b>x is greater</b>");
  }
  else
  {
    document.write("<b>y is greater</b>");
  }
</script>
</body>
</html>

```

**If else if**

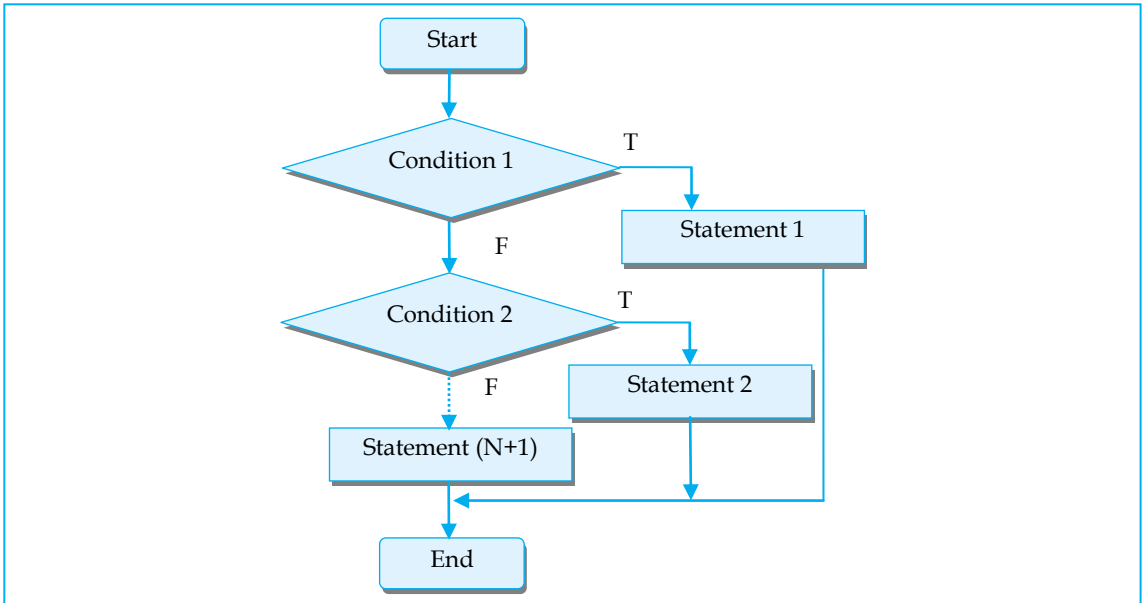
The **if...else if...** statement is an advanced form of **if...else** that allows JavaScript to make a correct decision out of several conditions.

**Syntax:**

```

if (condition 1)
{
  Statement(s) to be executed if condition 1 is true
}
else if (condition 2)
{
  Statement(s) to be executed if condition 2 is true
}
else if (condition 3)
{
  Statement(s) to be executed if condition 3 is true
}
else
{
  Statement(s) to be executed if no condition is true
}

```

**Flowchart:****Example:**

```

<html>
  <body>
    <script type = "text/javascript">
      var x=5,y=5;
      if(x>y)
      {
        document.write("X is greater");
      }
      else if(y>x)
      { document.write("y is greater");
      }
      else
      {
        document.write("x and y are equal");
      }
    </script>
  </body>
</html>

```

**Switch case**

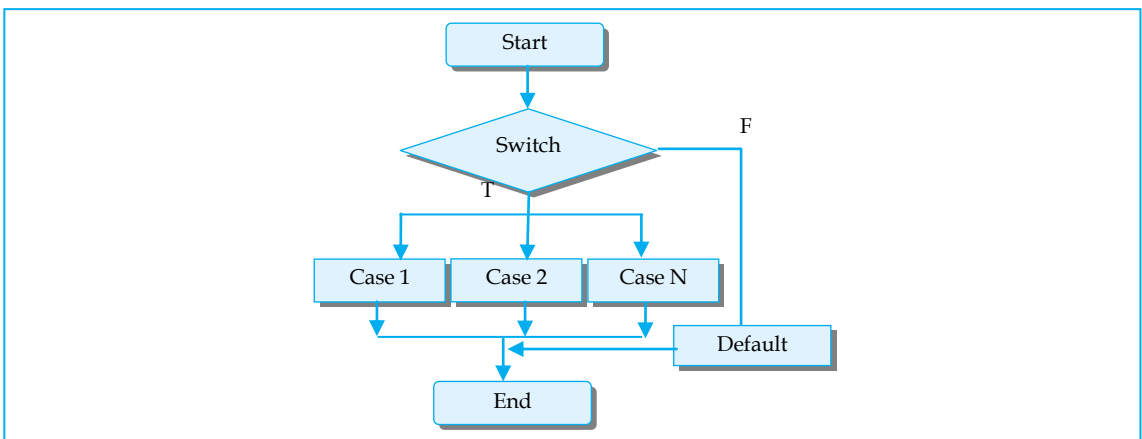
The switch statement is similar to the if-else statement. Instead of using multiple if else statements you can use switch case if there are many conditions to check.

**Syntax:**

```

switch(expression)
{
    case value1:
        statement1;
        break;
    case value2:
        statement2;
        break;
    default:
        default statement;
}

```

**Flowchart:****Example:**

```

<html>
<body>
    <script type = "text/javascript">
        var grade='B';
        switch (grade)
        {
            case 'A':
                document.write("Your Grade is A");
                break;
            case 'B':
                document.write("Your Grade is B");
                break;
            case 'C':
                document.write("Your Grade is C");
                break;
            case 'D':
                document.write("Your Grade is D");

```

```

        break;
    case 'F':
        document.write("Your Grade is F");
        break;
    default:
        document.write("Unknown Grade")
    }
</script>
</body>
</html>

```

## Looping Control Structure

Looping control structures are used when we want to execute a block of statement repeatedly. A loop executes the sequence of statements many times until some condition becomes false. There are different types of loop supported by JavaScript like other programming languages.

### For Loop

For loop is the commonly used loop. For loop is an entry control loop because it checks the condition at entry point. Hence, it does not execute even a single statement if the termination condition is set false.

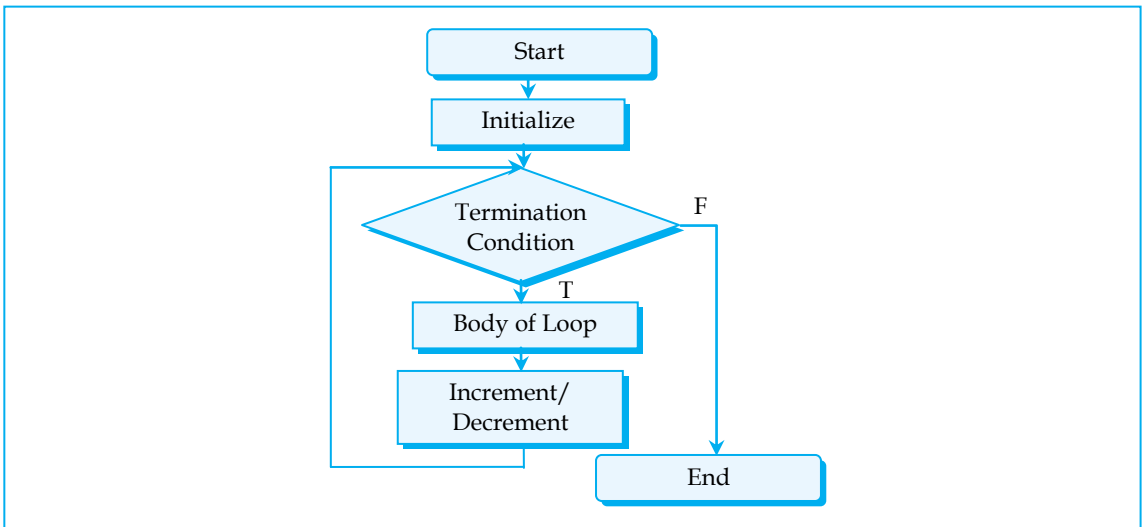
#### Syntax:

```

for (begin; condition; step)
{
    // body of loop
}

```

#### Flowchart:



**For example:**

```
for(let i = 0; i < 3; i++)
{
  // shows 0, then 1, then 2
  alert(i);
}
```

**Example explained:**

Looping parts	Statement	Explanation
begin	i = 0	Executes once upon entering the loop.
condition	i < 3	Checked before every loop iteration. If false, the loop stops.
step	i++	Executes after the body on each iteration but before the condition check.
body	alert(i)	Runs again and again while the condition is true.

Write a HTML and JavaScript code to display "Computer Science" 10 times using for loop

```
<html>
<body>
  <script type = "text/javascript">
    var i=1;
    for ( i; i < 10; i++)
      {
        document.write("Computer Science </br>");
      }
  </script>
</body>
</html>
```

**Result:**

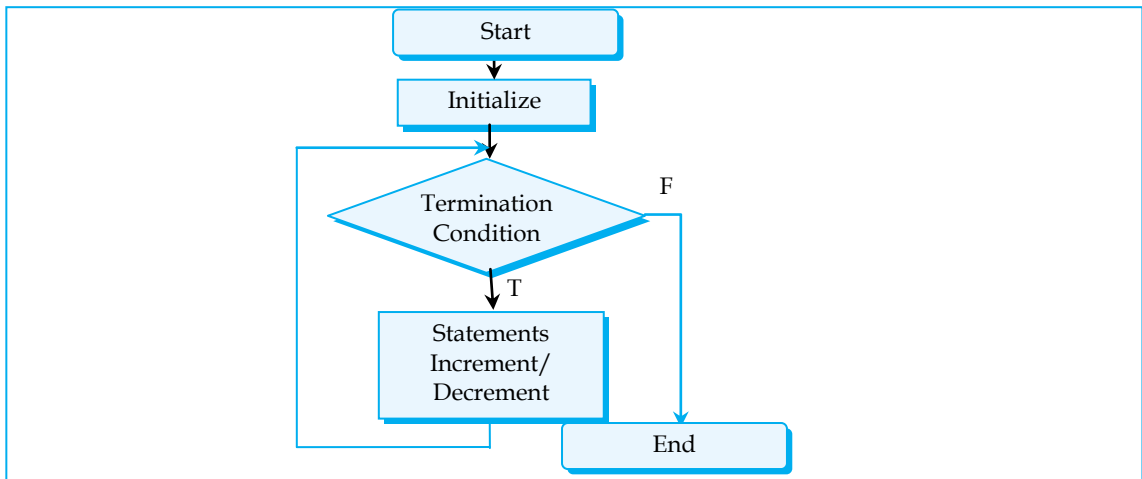
Computer Science  
 Computer Science  
 Computer Science  
 Computer Science  
 Computer Science  
 Computer Science  
 Computer Science  
 Computer Science  
 Computer Science

**While Loop**

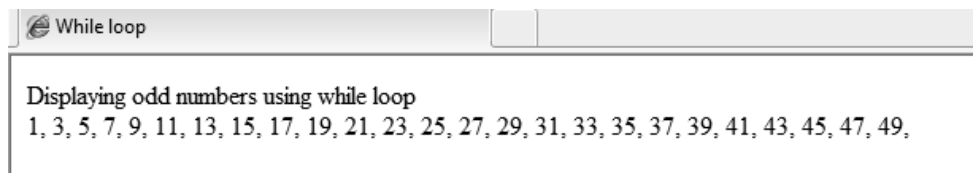
While loop is an entry controlled loop. In while loop, condition is evaluated first and if it returns true then the statements inside while loop execute. When condition returns false, the control comes out of loop and jumps to the next statement after while loop. When using while you need to use increment or decrement statement inside while loop so that the loop variable gets changed on each iteration and at some point condition returns false. This way you can end the execution of while loop otherwise the loop would execute indefinitely.

**Syntax:**

```
while (condition)
{
    statement (s);
    increment/decrement;
}
```

**Flowchart:****Example:**

```
<html>
  <title>While loop</title>
<body>
  Displaying odd numbers using while loop</br>
  <script type = "text/javascript">
    var i=1;
    while (i<=50)
    {
      document.write(i+", ");
      i=i+2;
    }
  </script>
</body>
</html>
```

**Result:**

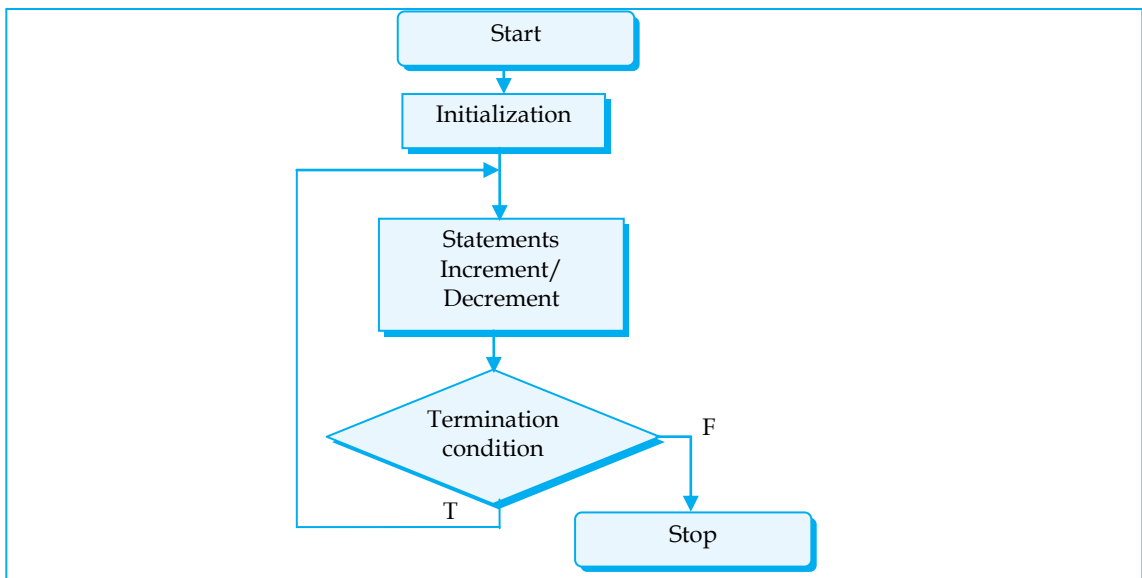
## Do While Loop

Do while loop is an exit controlled loop where loop condition is evaluated after the execution of loop's body. First, the statements inside loop execute and then the condition gets evaluated, if the condition is true then the control gets transferred to the "do" part of the loop. Otherwise it terminates the loop.

### Syntax:


```
do
{
    statement(s);
    increment/decrement;
}while(condition);
```

### Flowchart:



### Example:

```
<html>
  <title>Do While loop</title>
<body>
  Displaying even numbers using do while loop<br>
  <script type = "text/javascript">
    var i=2;
    do
    {
      document.write(i+", ");
      i=i+2;
    }while(i<=50);
  </script>
</body>
</html>
```

**Result:**


Displaying even numbers using do while loop

2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50.

**1. JavaScript Popup Boxes:**

JavaScript supports three important types of dialog boxes. These dialog boxes can be used to raise and alert, or to get confirmation on any input or to have a kind of input from the users.

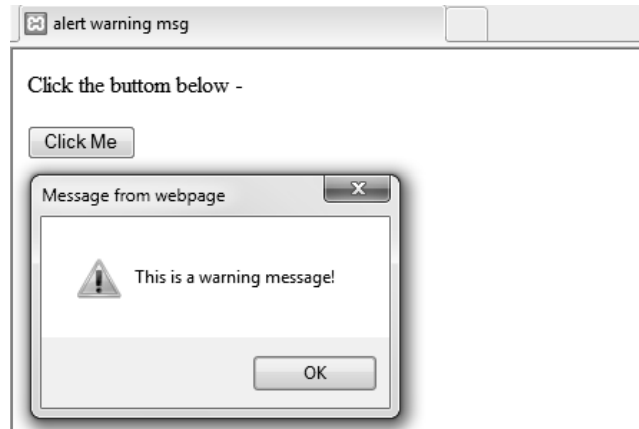
**Alert Dialog Box**

An alert dialog box is mostly used to give a warning message to the users. For example, if one input field requires to enter some text but the user does not provide any input, then as a part of validation, you can use an alert box to give a warning message.

Nonetheless, an alert box can still be used for friendlier messages. Alert box gives only one button "OK" to select and proceed.

**Example:**

```
<html>
  <title>alert warning msg</title>
  <head>
    <script type="text/javascript">
      function Warn()
      {
        alert ("This is a warning message!");
        document.write ("This is a warning message!");
      }
    </script>
  </head>
  <body>
    Click the button below -</br>
    <form>
      <input type="button" value="Click Me" onclick="Warn();" />
    </form>
  </body>
</html>
```

**Result:****2. Confirmation Dialog Box**

A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: Ok and Cancel.

If the user clicks on the OK button, the window method `confirm ()` will return true. If the user clicks on the Cancel button, then `confirm ()` returns false. You can use a confirmation dialog box as follows.

```
<html>
  <title>Conform warning msg</title>
  <script type="text/javascript">
    function getConfirmation()
    {
      var retVal = confirm("Do you want to continue ?");
      if( retVal == true ){
        document.write ("User wants to continue!");
        return true;
      }
      else{
        Document.write ("User does not want to continue!");
        return false;
      }
    }
  </script>

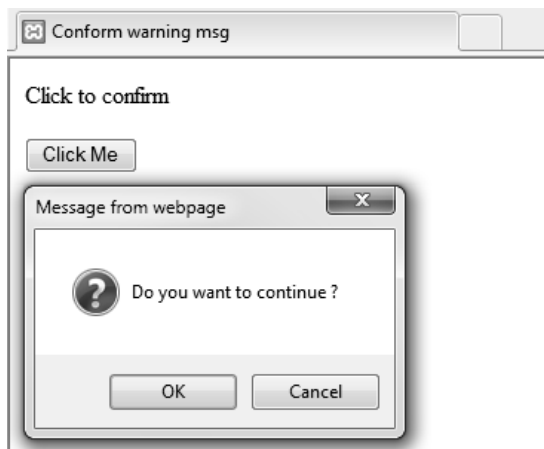
</head>
<body>
  Click to confirm<br/>
  <form>
    <input type="button" value="Click Me" onclick="getConfirmation();" />
  </form>
</body>
</html>
```

```
</form>
```

```
</body>
```

```
</html>
```

Result:



### 3. Prompt Dialog Box

The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus, it enables you to interact with the user. The user needs to fill in the field and then click OK.

This dialog box is displayed using a method called `prompt()` which takes two parameters:

- a label which you want to display in the text box and
- a default string to display in the text box.

This dialog box has two buttons: OK and Cancel. If the user clicks the OK button, the window method `prompt()` will return the entered value from the text box. If the user clicks the Cancel button, the window method `prompt()` returns null.

#### Example:

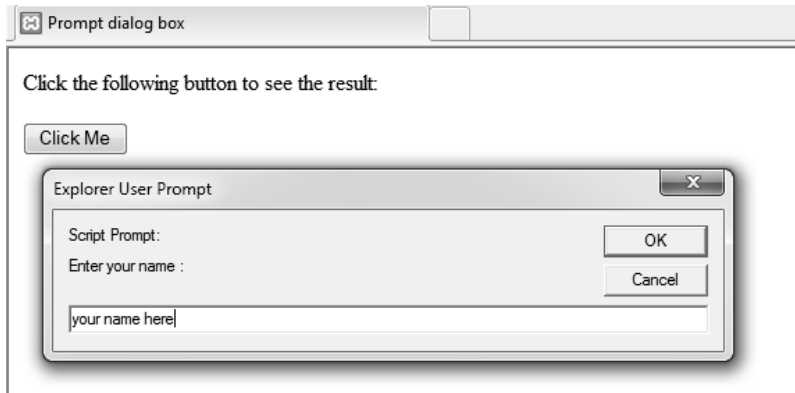
```
<html>
  <head>
    <title>Prompt dialog box</title>
    <script type="text/javascript">
      function getValue()
      {
        var retVal = prompt("Enter your name : ", "your name here");
        document.write("You have entered : " + retVal);
      }
    </script>
  </head>
```

```

<body>
  <p>Click the following button to see the result: </p>

  <form>
    <input type="button" value="Click Me" onclick="getValue();" />
  </form>
</body>
</html>

```

**Result:**

## Some Solved Examples

### 1. Write a JavaScript and HTML code to display the Fibonacci series up to 20

```

<html>
<body>
  <script type = "text/javascript">
    var a = 0, b = 1, c;
    document.write("Fibonacci series <br />");
    while(a<20)
    {
      document.write(a + ", ");
      c = a+b;
      a = b;
      b = c;
    }
  </script>
</body>
</html>

```

**Result:**

**Fibonacci series**

**0, 1, 1, 2, 3, 5, 8, 13**

## 2. Write a JavaScript program to display even numbers less than 50 using do while loop

```
<html>
<body>
  Even numbers less than 50<br/>
  <script type = "text/javascript">
    var i = 2;
    do
    {
      document.write(i + ", ");
      i = i + 2;
    }while(i<50);
  </script>
</body>
</html>
```

### Result:

#### Even numbers less than 50

2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48,

## 3. Write a JavaScript and HTML to ask a number from a user and display the name of day according to user's choice.

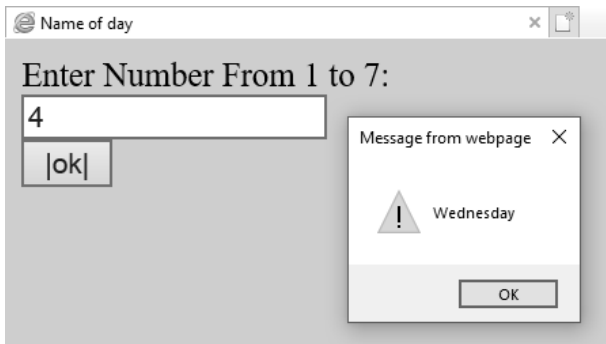
```
<html>
<head>
  <title>
    Name of day
  </title>
<script type="text/JavaScript">
function calculate()
{
  var n=document.getElementById("number").value;
  switch(n)
  {
    case '1':
      alert("Sunday");
      break;
    case '2':
      alert("Monday");
      break;
    case '3':
      alert("Tuesday");
      break;
    case '4':
      alert("Wednesday");
      break;
    case '5':
```

```

        alert("Thursday");
        break;
    case '6':
        alert("Fridayday");
        break;
    case '7':
        alert("Saturdy");
        break;
    default:
        alert("Wrong Choice !");
    }
}
</script>
</head>
<body bgcolor="pink" >
<form name="day">
Enter Number From 1 to 7: <br>
<input type="text" id="number"/><br/>
<input type="submit" value="|ok|" onclick="calculate()"/>
</form>
</body>
</html>

```

Result:



#### 4. Write a JavaScript code to enter two numbers and display their sum and product

```

<html>
<head>
    <title>Arithmetic Calculation</title>
<script type="text/JavaScript">
function add()
{
    var a,b,c;
a=Number(document.getElementById("first").value);
b=Number(document.getElementById("second").value);
c= a + b;

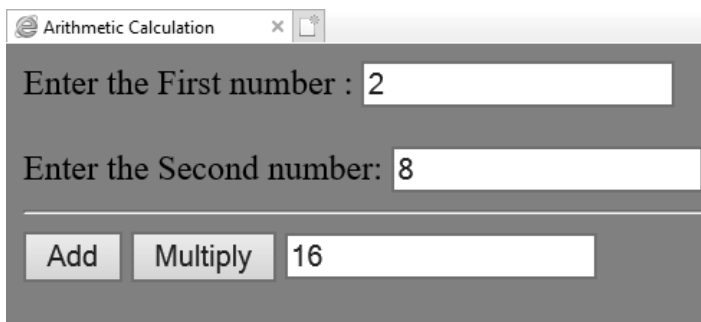
```

```

document.getElementById("answer").value= c;
}
function multiply()
{
    var a,b,c;
a=Number(document.getElementById("first").value);
b=Number(document.getElementById("second").value);
c= a * b;
document.getElementById("answer").value= c;
}
</script>
</head>
<body bgcolor="DodgerBlue">
Enter the First number : <input id="first"> <br><br/>
Enter the Second number: <input id="second"> <br><hr/>
<button onclick="add()">Add</button>
<button onclick="multiply()">Multiply </button>
<input id="answer">

</body>
</html>

```

**Result:****Exercise****Short Answer Questions**

1. What do you mean by scripting language? Describe the types of scripting language.
2. Differentiate between static and dynamic web pages.
3. Differentiate between server side and client side scripting.
4. Differentiate between programming language and scripting language.
5. Difference between while and do while loop.
6. What do you mean by JavaScript? Write the advantages of using JavaScript.
7. Explain how to add JavaScript code to HTML.

8. Describe the method to use function in JavaScript with example.
9. Explain the data types used in JavaScript.
10. What is variable? How variable is declared in JavaScript?
11. What is operator? Explain the types of operator used in JavaScript.
12. Explain increment and decrement operator with example.
13. What is loop? Explain types of loop with example.
14. What is event? Describe the types of events in HTML
15. What is function? Write its advantages.
16. What are the benefits of using JavaScript in a HTML page?

### ■ Programming Questions

1. Write a JavaScript and HTML code to input cost price and sales price and find the profit or loss.
2. Write a JavaScript and HTML code to display the following series: 1, 2, 3, 4, 5, 6 ..... 100.
3. Write a JavaScript and HTML code to display the following series: 1, 4, 9, 16, 25, ..... up to 10<sup>th</sup> term.
4. Write a JavaScript and HTML code to enter a number and display whether it is prime or not.
5. Write a JavaScript and HTML code to display the multiplication table of a given number.
6. Write a JavaScript and HTML code to display the following patterns:
  - a. 

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```
  - b. 

```
1
1 1
1 1 1
1 1 1 1
1 1 1 1 1
```
7. Write a JavaScript and HTML code to input a number and display whether it is odd or even
8. Write a JavaScript and HTML code display the sum of 'n' terms of odd numbers.
9. Write a JavaScript program to display simple interest.
10. Write a JavaScript program to display the area of rectangle
11. Write a JavaScript program to display the factorial of a numbers using function
12. Write a JavaScript and HTML code to display all the numbers from 1 to 10.
13. Write a JavaScript and HTML code to display the multiplication of a number.
14. Write a JavaScript and HTML code to display all even numbers up to 50
15. Write a JavaScript function to Fahrenheit that converts a temperature from Celsius to Fahrenheit. [Fahrenheit= 1.8 \* Celsius + 32]
16. Write a JavaScript function product that takes two numbers and returns their product.
17. Write a HTML and JavaScript code to enter a number from a user and display whether it is positive or negative.

## Object Based Programming with JavaScript and Event Handling

### Event Handling

Event can be something the browser does, or something a user does. JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page. When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, dragging a mouse, moving a mouse, selecting a text, etc. Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, etc. Often, when events happen, you may want to do something.

Some common events are:

Event type	Attributes	Descriptions
<b>Windows Events</b>	Onload	Event occurs when the webpage is entirely loaded.
	Onunload	Event occurs when the current webpage is unloaded, or window is closed.
	Onresize	Event occurs when the window is resized.
	Onoffline	Event occurs when the browser started working offline
	Ononline	Event occurs when the browser started working online
<b>Form Events</b>	Onchange	Event occurs when the value of the element is changed.
	Onfocus	Event occurs when the element gets focused.
	Onselect	Event occurs when the user has selected some text.
	Onsubmit	Triggers the event when a form is submitted.
	Onblur	Event occurs when form element loses the focus.
<b>Keyboard Events</b>	Onkeydown	Triggers the event when the user presses down a key on the keyboard.
	Onkeypress	Trigger the event when the user presses any key on the keyboard.
	Onkeyup	Trigger the event when the user releases the currently pressed key.
<b>Mouse Events</b>	OnClick	Triggers the event when the mouse clicks on the element.
	Ondblclick	Triggers the event when mouse double-click occurs on the element.
	onmousedown	Triggers the event when the mouse pointer moves over the element.
	onmouseover	Triggers the event when the mouse moves onto the element.
	Onmouseup	Triggers the event when the mouse button is released.
	onmousemove	Triggers the event when the mouse pointer moves over the element.

### OnClick Event:

This is the most frequently used event type which occurs when a user clicks the left button of his mouse. You can put your validation, warning etc., against this event type.

**Example:**

```
<html>
<head>
<script type="text/javascript">
    function sayHello()
```

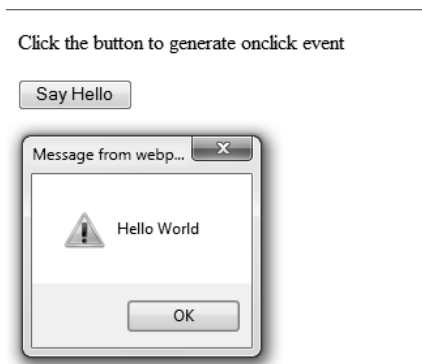
```

    {
      alert("Hello World");
    }
  </script>
</head>
<body>
Click the button to generate onclick event
<form>
<input type="button"onclick="sayHello()"value="Say Hello"/>
</form>

</body>
</html>

```

Result:



## Onfocus/onfocusout Event

The onfocus event occurs when an element gets focus. The onfocusout event occurs when an element is about to lose focus.

### Example:

```

<html>
  <title>On focus Event</title>
<body>
  <p>Click on text box to call focus event</p>
  <p>Click on anywhere to to call unfocusout event</p>

```

Enter your name:

```

<input type="text" id="fname" onfocus="change()" onfocusout="notchanged(
)">

```

```

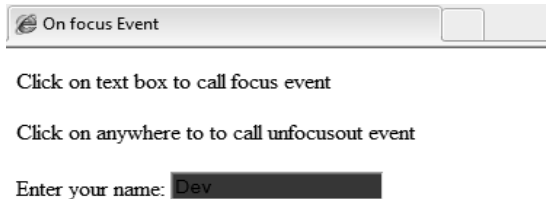
<script>
function change()
{
  document.getElementById("fname").style.backgroundColor = "yellow";
}
function notchanged()

```

```

{
  document.getElementById("fname").style.backgroundColor = "red";
}
</script>
</body>
</html>

```

**Result:****OnKeyUp Event**

This event is generated by keyboard. It triggers when a key is released. The example below illustrates the concept of onkeyup event.

```

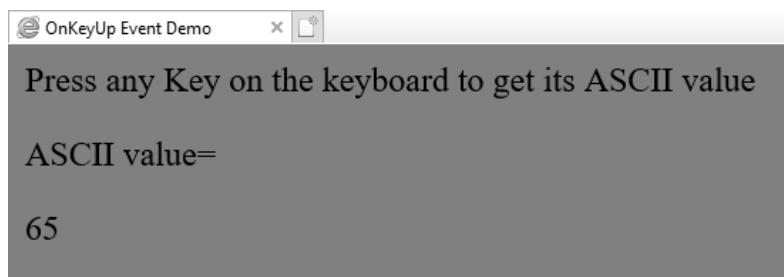
<html>
<head>
  <title>OnKeyUp Event Demo</title>
<script type="text/JavaScript">
function Button(event)
{
  document.getElementById("value").innerHTML = event.keyCode;
}
</script>
</head>

<body onkeyup="Button(event)">
<p>Press any Key on the keyboard to get its ASCII value </p>

ASCII value= <p id="value"></p>

</body>
</html>

```

**Result:**

## Image and form Objects

HTML forms are used to take input from users. For example, to create an account of any user you need to collect information such as: name, address, phone number, etc. To collect data from a user HTML <form> tag is used. A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

The HTML <form> tag is used to create an HTML form and it has following syntax:

```
<form action = "Script URL" method = "GET|POST">
    form elements like input, textarea etc.
</form>
```

You have learnt different form elements in class XI. Some common form elements are:

Tag	Attributes	What it does		
Form		Starts a form on web page.		
Input	Type	Specifies the type of form input to accept different types of values from the user.		
		Value for the type attribute Type= "text" Type= "password" Type= "radio" Type= "checkbox" Type="button" Type= "submit"	Type of the form object To insert textbox object on the form. To insert password entry box object on the form. To insert radio button object on the form. To insert check box object on the form. To insert normal button To insert submit button on the form.	
	Name	Specifies the name of the object of the form.		
	Size	Specifies the length of the text entry field, in terms of number of characters. It is applied only for text and password object.		
	Maxsize	Specifies the maximum length of the text entry field, in term of number of character.		
	Select		Enables the user to select one or more items from a menu or a scrolling list. They are similar in functionality to radio or check box, but they are display in a different way on the screen.	
		Name	Specifies the name of the select list.	
	Multiple	Specifies the user can select multiple options from the list.		
Option		Specifies the individual option list within the selection list. It is written inside the <select> tag.		
Text area		To insert textarea in which the user can type many lines of text such as email, sms, comment etc.		
	Name	Specifies the name of the text area.		
	Rows	Specifies the height of the text area object, in term of rows of text.		
	Cols	Specifies the width of the text area object, in term of columns (characters).		

## Form Validation

In an interactive website, client interacts with server by entering some data through the HTML forms. Form validation is the method to check if the data entered by client is missing or incorrect. Form validation can be done on server after submitting data or on the client browser

before submitting data to the server. You can use JavaScript to validate user's data on client's browser before it is send to the server. To validate user's data you must know how to take the values entered by the user in a form.

You can validate your form against the following:

- Checking for empty text boxes
- Numbers validation
- Date validation
- Check for letters
- Check the selection made or not
- Email validation
- Password Validation
- Check the length of the input text

There are two methods for validating forms: server-side (PHP, ASP, etc), and client-side (usually JavaScript). Server-side validation is more secure but more difficult to code, whereas client-side validation is easier. The browser doesn't have to connect to the server to validate the form, so the user finds out instantly if they've missed out that required field.

Here we validate the form data using JavaScript. To validate the user data, you have to get the data/ value entered by the user through form.

To get the value following syntax is used.

### Syntax:

```
var x = document.getElementById("myText").value;  
or  
var x = document.FormName.TextFieldName.value;
```

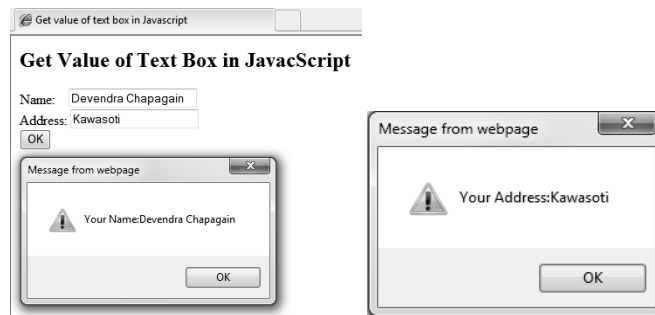
### Example:

```
<html>  
<head>  
<title>Get value of text box in Javascript</title>  
  
<script type="text/javascript">  
function GetValue()  
{  
    var name = document.getElementById('txtUserName').value;  
    var address=document.MyForm.txtaddress.value;  
    alert("Your Name:"+name);  
    alert("Your Address:"+address);  
}  
</script>  
</head>  
<body>
```

```

<h2>Get Value of Text Box in JavaScript</h2>
<form name="MyForm">
Name:   &nbsp; &nbsp; &nbsp;<input id="txtUserName" type="text" /><br/>
Address: <input id="txtaddress" type="text" name="txtUser"/><br/>
<input type="button" value=" OK " onclick="GetValue()" />
</form>
</body>
</html>

```

**Result:**

An example of simple form validation using JavaScript is given below:

```

<html>
<title>Form validation</title>
<head>
<title>Form Validation</title>
<script type="text/javascript">
// Form validation code will come here.

var name=document.myForm.Name.value;
var email=document.myForm.EMail.value;
var zip=document.myForm.Zip.value;
var country=document.myForm.Country.value;
function validate()
{
if(name=="")
{
    alert("Please provide your name!");
    document.myForm.Name.focus();
return false;
}

if(email=="")
{
    alert("Please provide your Email!");

```

```

        document.myForm.Email.focus();
return false;
}

if(zip==""||isNaN(zip)||zip.length !=5)
{
    alert("Please provide a zip in the format #####.");
    document.myForm.Zip.focus();
return false;
}

if(country =="-1")
{
    alert("Please provide your country!");
return false;
}
return(true);
}
</script>
</head>

<body>
<form name="myForm"onsubmit="return(validate());">
<table cellpadding="2" cellspacing="2" border="1">

<tr>
<td align="right">Name</td>
<td><input type="text" name="Name" placeholder="Enter Name"/></td>
</tr>
<tr>
<td align="right">EMail</td>
<td><input type="text" name="EMail"/></td>
</tr>
<tr>
<td align="right">Zip Code</td>
<td><input type="text" name="Zip"/></td>
</tr>

<tr>
<td align="right">Country</td>
<td>
<select name="Country">
    <option value="-1"selected>[choose yours]</option>

```

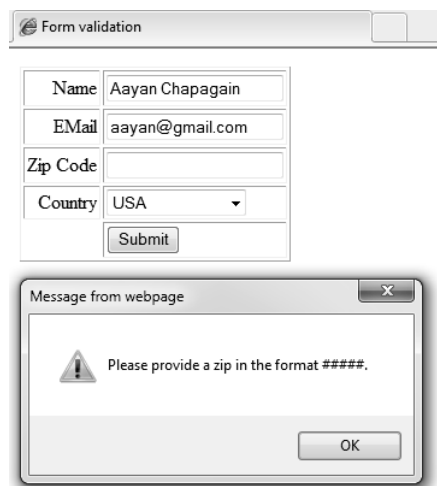
```

    <option value="1">USA</option>
    <option value="2">UK</option>
    <option value="3">INDIA</option>
</select>
</td>
</tr>
<tr>
    <td align="right"></td>
    <td><input type="submit" value="Submit"/></td>
</tr>
</table>
</form>

</body>
</html>

```

### Result:



### JQuery:

jQuery is a small JavaScript library that makes development of HTML-based client JavaScript significantly easier. jQuery is a fast, small, and feature-rich JavaScript library. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for Rapid Web Development.

- jQuery is a JavaScript Library.
- jQuery greatly simplifies JavaScript programming.
- JQuery Motto: Write less, do more.
- jQuery is a framework built using JavaScript capabilities
- jQuery helps to add animation and effects easily

There are two ways to include it in your project:

- Download a local copy
- Link to the file via Content Delivery Network (CDN)

## Exercise

1. What do you mean by event handling? Explain any two events with example
2. What is jQuery? Why it is used in the web development?

## ■ Programming Exercise

1. Write a HTML code to display the following registration form and validate the inputs using JavaScript.

The validation must include the following

- Input fields user id, password, name and country must not be blank
- Email must be valid email
- User id must be integer number not more than 3 digits
- Password length cannot be less than 5 characters and more than 8 characters

### Registration Form

<b>User id:</b>	<input type="text" value="101"/>
<b>Password:</b>	<input type="password" value="*****"/>
<b>Name:</b>	<input type="text" value="Devendra Chapagain"/>
<b>Address:</b>	<input type="text" value="Kawasoti-8, NP"/>
<b>Country:</b>	<input type="text" value="Nepal"/>
<b>ZIP Code:</b>	<input type="text" value="+977"/>
<b>Email:</b>	<input type="text" value="devcpgn@gmail.com"/>
<b>Gender:</b>	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Female
<b>Language:</b>	<input checked="" type="checkbox"/> English <input type="checkbox"/> Non English
<b>About:</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">         DEVENDRA CHAPAGAIN obtained his Masters Degree in Computer Application from Lovely Professional University (LPU) during 2014 and Bachelors degree from Pokhara University, Pokhara, Nepal during 2010 and He specialized on database and Data warehouse as well as specialization on ASP.Net . His research area includes data warehousing and data mining. He has 8+ years experience of teaching for Bachelor levels and +2 levels. He love to design websites using HTML,CSS, ASP.NET,PHP,Jquery, Javascript. Currently he is lecturer at Birendra Multiple College,Bhatatpur, Chitwan and Oxford College of Engineering and Management, Siddhartha, Nepal.       </div>
<input type="button" value="Submit"/>	

## Server Side Scripting with PHP

Server side scripting runs on the server. Server side scripting is used to retrieve and generate dynamic content. The commonly used server side scripting is: PHP, Ruby, Ruby on Rails, ASP, etc. One disadvantages of server side scripting is that scripting software must be installed on the content management system tools in order to store the dynamic data.

## Common Server Side Scripting Languages

- **ASP/ASP.net** –Active Server Pages developed by Microsoft to make advanced web pages. Unix servers use Chillisoft ASP. Allows user to get any .net-enabled language to program a site.
- **JSP/Servlets** – Java Server Pages which include JSP tags mixed in with html. Most popular language for higher level applications.
- **ColdFusion** – Runs on top of a JSP/servlet engine. Can cost \$1,299 a server. Users can download a free “developer” edition but that is limited to one IP address.
- **PHP**- PHP is easy to learn, if you are comfortable with C

After a website is developed, it must be uploaded on a web server, to make it available for other users to see. The web server’s responses to the client requests and display the web pages of the web site to the user. If the web server is down, then the web site will not load or displayed. There are many different types of web server provided by different companies. You can choose any of them according to your requirement. The commonly used web servers in the market are:

- Apache HTTP Server
- Microsoft Internet Information Services
- Lighttpd
- Nginx Web Server
- Sun Java System Web Server

## Introduction to PHP: Hardware and Software Requirement

### Introduction to PHP

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. The product was initially named *Personal Home Page*. It is used to create web applications in combining with a web server, such as Apache. PHP has nothing to do with layout, events, and nothing about the look and feel of a web page. In fact most of what PHP does is invisible to the user.

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use
- PHP 8 is used currently
- PHP is free to use
- PHP is easy to learn and write
- Returns simple output to the client
- PHP files have a file extension of ".php"
- PHP runs on different platforms (Windows, Linux, Unix, etc.)

**Note:** To run PHP you need to install XAMPP server and configure it. The XAMPP server configuration is described later in the chapter.

## Object Oriented Programming with Server Side Scripting

### Object Oriented Programming in PHP

Object-Oriented Programming (OOP) is a programming model that is based on the concept of classes and objects. As opposed to procedural programming where the focus is on writing procedures or functions that perform operations on the data, in object-oriented programming the focus is on the creations of objects which contain both data and functions together.

Object-oriented programming has several advantages over conventional or procedural style of programming. The most important ones are listed below:

- It provides a clear modular structure for the programs.
- It is easier to understand because code written with the concept of OOP is very structured.
- It makes your code much easier to maintain, modify and debug.
- It provides high degree of reusability. The concept of reusability allows writing a code once and calling many times in the program.
- It reduces the development time with less line of code (LOC).
- You can hide the program complexity
- You can develop complex and reusable web applications
- Secure program can be developed
- It provides many features like: inheritance, object, class, polymorphism, encapsulation, abstraction etc.

Object-oriented concept of PHP is same as other programming like C++ and java programming. The following are the list of terms that are used in creating object oriented applications in PHP.

- **Class:** This is a programmer-defined data type, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or class) of object.
- **Object:** An individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.
- **Inheritance:** When a class is defined by inheriting existing function of a parent class then it is called inheritance. Here child class will inherit all or few member functions and variables of a parent class.
- **Polymorphism:** This is an object oriented concept where same function can be used for different purposes. For example function name will remain same but it make take different number of arguments and can do different task.

- **Overloading:** overloading is a type of polymorphism in which some or all of operators have different implementations depending on the types of their arguments. Similarly functions can also be overloaded with different implementation.
- **Data Abstraction:** Any representation of data in which the implementation details are hidden (abstracted).
- **Encapsulation:** It refers to a concept where we encapsulate all the data and member functions together to form an object. This concept of encapsulation allows hiding of unnecessary details in a program.

## Basic PHP Syntax

Creating a PHP file is similar to creating an HTML file. In fact, most PHP files are a mixture of PHP code and HTML. To create a PHP file, simply do the following:

- Create a new file in your favorite editor
- Type some PHP code
- Save the file with a .php extension

A PHP script can be placed anywhere in the document.

A PHP script starts with `<?php` and ends with `?>`

```
<?php
  // PHP code goes here
?>
```

The default file extension for PHP files is ".php".

A PHP file normally contains HTML tags, and some PHP scripting code.

### Example:

```
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "Hello World!";
?>
</body>
</html>
```



## My first PHP page

Hello World!

## Steps to run a PHP program

- Step 1:** To write a code you need a text editor. Some of the commonly used editors are: Sublime Text, Notepad++, Visual Code Studio, CoffeeCup, EditPlus, gedit, Dreamweaver etc. Download your favorite text editor.
- Step 2:** PHP requires a Server to run the code. Therefore download the web server. PHP runs fine on Apache Web server. The steps of installing the XAMPP server are illustrated later in the chapter. See Steps Involved in Making Connection with MySQL Database. Run the XAMPP server before running your code.
- Step 3:** Start the Apache and MySQL services from XAMPP control pannel
- Step 4:** Write a PHP code in your favorite editor.
- Step 5:** Save the file in location: C:\xampp\htdocs\Filename.php
- Step 6:** Open the web browser and type: http://localhost/Filename.php/
- Step 7:** You are done! Your code runs on the server.
- Step 8:** Enjoy your PHP program

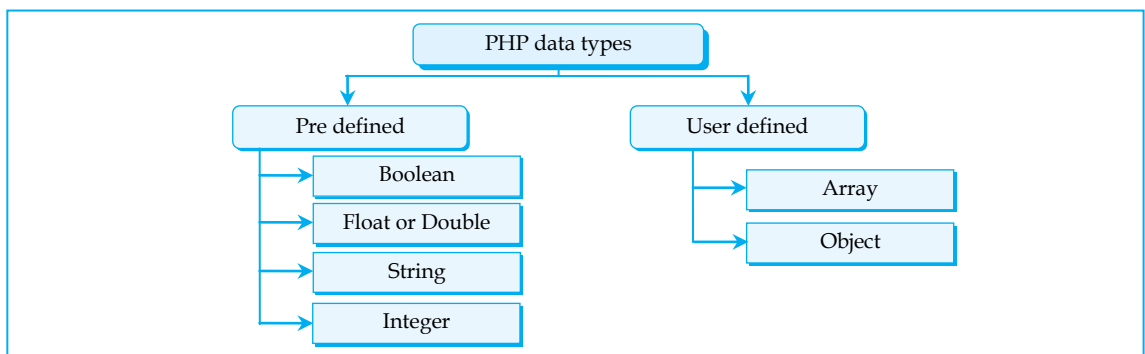
## Using Phpinfo() Function

phpinfo() is a PHP built-in function used to display information about PHP's configuration settings and modules.

```
<?php
phpinfo();
?>
```

## PHP Data Types

Data Types defines the type of data a variable can store. Unlike other programming languages PHP also has different data type. However it is not compulsory to define the data type during declaration of variable. PHP automatically converts the variable to the correct data type, depending on its value. The different data types in PHP are:



## PHP Booleans

Booleans has only two possible values either TRUE (1) or FALSE (0). Booleans are usually used in conditional testing statements.

## PHP Integers

Integer is a numeric data type without decimal point (i.e. 1, 5, 35,-6, etc.). Integers can be either positive or negative. It can also be decimal, hexadecimal, or octal. The default is decimal.

## PHP Float or Double

Floating point numbers also known as "floats", "doubles", or "real numbers" are decimal or fractional numbers. For example: 3.5, 2.9, -10.6, 4E5 etc.

## PHP Strings

Strings can be enclosed in either single or double quote with different behavior at read time.

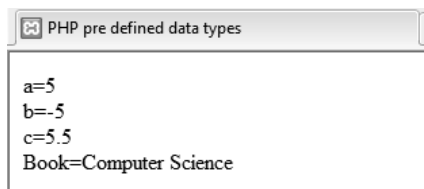
**Singly quoted string:** Singly quoted strings read in and store their characters literally.

**Doubly Quoted String:** Variable names (string with \$) are replaced with string representation of their values.

Program to illustrate the concepts of different data types in PHP

```
<html>
  <head>
    <title>PHP pre defined data types </title>
  </head>
  <body>
    <?php
    $a = 5; // decimal number
    $b = -5; // a negative number
    $c = 5.5; // float or double
    $book="Computer Science"; // string
    echo "a=$a <br/> b=$b </br/>";
    echo "c=$c <br/> Book=$book";
    ?>
  </body>
</html>
```

**Result:**



```
PHP pre defined data types
a=5
b=-5
c=5.5
Book=Computer Science
```

## Basic Programming in PHP

PHP is web development server side scripting language, usually used to create web applications in combination with web server, such as Apache. PHP has nothing to do with layout, events and anything about the looks of the web page. The PHP as a server side scripting is used along with the static HTML and common client side technologies. PHP can interact with JavaScript, Java applets, Flash, Style sheets, and so on. The most basic type of web page is completely static written in HTML. You can add CSS (Cascading style sheet), JavaScript, or other client side technologies to make it interactive. The addition of client side technology is not enough for dynamic web sites. They depend entirely on browser. The need of server side scripting arises to connect web sites to backend servers for processing data.

PHP runs on various platforms, including Linux, Microsoft windows, and Mac OS. PHP is combined with Apache Web server and MySQL. Anything compatible with HTML is also compatible with PHP so it will be easier for you to learn PHP if you already know HTML.

## Variables Manipulation

### Variables

Variables are "containers" for storing information. PHP variable names are case-sensitive. PHP is a Loosely Typed Language. PHP automatically converts the variable to the correct data type, depending on its value. In other languages such as C, C++, and Java, the programmer must declare the name and type of the variable before using it.

In PHP, a variable starts with the \$ sign, followed by the name of the variable:

#### Example:

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.

#### Rules for PHP Variables:

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_)
- Variable names are case-sensitive (\$age and \$AGE are two different variables)

## Output Variables

### PHP Echo and Print Statements

The PHP echo and print statement is often used to output data to the screen. echo and print are more or less the same. They are both used to output data to the screen.

The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

```
<?php
$str = " Dev!";
$x = 5;
$y = 4;
echo $x + $y;
print $str;
print "<br>What a nice day!";
echo "I love $str!";
?>
```

## Operators

Operators are used to perform operations on variables and values. The operators used in PHP and other programming languages including JavaScript are similar. PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators
- Conditional Operators

### Arithmetic Operator

The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name	Example	Result
+	Addition	$\$x + \$y$	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \$y$	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \$y$	Product of $\$x$ and $\$y$
/	Division	$\$x / \$y$	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \$y$	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \$y$	Result of raising $\$x$ to the $\$y$ 'th power (Introduced in PHP 5.6)

**Example:**

```

<html>
  <head>
    <title>arithmetic operator</title>
  </head>
  <body>
    Arithmetic Operator <br/>
  <?php
    $x = 10;
    $y = 6;
    $s=$x+$y;// arithmetic addition
    $dif=$x-$y; //arithmetic subtraction
    echo "x=$x, y=$y<br>";
    echo "x+y=$s <br>";
    echo "x-y=$dif<br>";
    ?>
  </body>
</html>

```

**Result:**

**Arithmetic Operator**

x=10, y=6

x+y=16

x-y=4

**PHP Assignment Operators**

The PHP assignment operators are used with numeric values to write a value to a variable.

The basic assignment operator in PHP is "=". It means that the left operand gets set to the value of the assignment expression on the right.

Assignment	Same as	Description
$x = y$	$\$x = \$y$	The left operand gets set to the value of the expression on the right
$x += y$	$\$x = \$x + \$y$	Add and assign
$x -= y$	$\$x = \$x - \$y$	Subtraction and assign
$x *= y$	$\$x = \$x * \$y$	Multiplication and assign
$x /= y$	$\$x = \$x / \$y$	Division and assign
$x \% = y$	$\$x = \$x \% \$y$	Modulus and assign

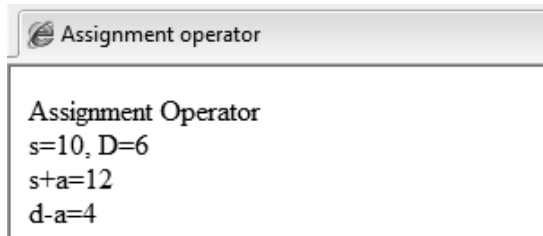
**Example:**

```

<html>
  <head>
    <title>Assignment operator</title>
  </head>
  <body>
    Assignment Operator <br/>
    <?php
    $s = 10;
    $d = 6;
    $a=2;
    echo "s=$s, D=$d<br>";
    $s+=$a; //same as s=s+a
    $d-=$a; // same as d=d-a
    echo "s+a=$s <br>";
    echo "d-a=$d<br>";
    ?>
  </body>
</html>

```

Result:



## PHP Relational Operators

The PHP relational operators are used to compare two or more values. The result of PHP relational operator is always either TRUE or FALSE. The common operators are:

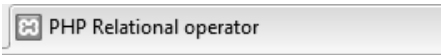
Operator	Name	Example	Result
==	Equal	\$x == \$y	Returns true if \$x is equal to \$y
===	Identical	\$x === \$y	Returns true if \$x is equal to \$y, and they are of the same type
!=	Not equal	\$x != \$y	Returns true if \$x is not equal to \$y
<>	Not equal	\$x <> \$y	Returns true if \$x is not equal to \$y
!==	Not identical	\$x !== \$y	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	\$x > \$y	Returns true if \$x is greater than \$y
<	Less than	\$x < \$y	Returns true if \$x is less than \$y
>=	Greater than or equal to	\$x >= \$y	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	\$x <= \$y	Returns true if \$x is less than or equal to \$y

**Example:**

```

<html>
  <head>
    <title>PHP Relational operator</title>
  </head>
  <body>
    PHP Relational Operator <br/>
    <?php
    $x=6; $y=6;
    $a=2; $b=3;
    if($x==$y)
    echo"$x and $y are equal <br/>";
    if($a!=$b)
    echo "$a and $b are not equal"
    ?>
  </body>
</html>

```

**Result:**


**PHP Relational Operator**  
 6 and 6 are equal  
 2 and 3 are not equal

**PHP Increment/Decrement Operators**

The PHP increment operators are used to increment a variable's value by 1.

The PHP decrement operators are used to decrement a variable's value by 1.

Operator	Name	Description
++\$x	Pre-increment	Increments \$x by one, then returns \$x
\$x++	Post-increment	Returns \$x, then increments \$x by one
--\$x	Pre-decrement	Decrements \$x by one, then returns \$x
\$x--	Post-decrement	Returns \$x, then decrements \$x by one

**Example:**

```

<html>
  <head>
    <title>Inc/Dec operator</title>
  </head>
  <body>
    <b> PHP Inc/Dec Operator</b> <br/>
    <?php

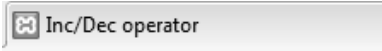
```

```

$x = 10;
echo "x=$x ";
$x++;
echo "After x++=$x <br/>";
$y = 5;
echo "y=$y ";
--$y;
echo "After --y=$y";

?>
</body>
</html>

```

**Result:****PHP Inc/Dec Operator**

x=10 After x++=11

y=5 After --y=4

**PHP Logical Operators**

The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	\$x and \$y	True if both \$x and \$y are true
or	Or	\$x or \$y	True if either \$x or \$y is true
xor	Xor	\$x xor \$y	True if either \$x or \$y is true, but not both
&&	And	\$x && \$y	True if both \$x and \$y are true
	OR	\$x    \$y	True if either \$x or \$y is true
!	NOT	!\$x	True if \$x is not true

**Example:**

```

<html>
  <head>
    <title>Logical operator</title>
  </head>
  <body>
    <b> PHP Logical Operator</b> <br/>
    <?php
      $class = 12; $subject = "Computer";


      if($class==12 AND $subject=="Computer")
        echo "Welcome to PHP <br/>";

```

```

        if($class==12 OR $subject=="Computer")
        echo "Welcome to class 12 ";
        ?>
    </body>
</html>

```

**Result:**

**PHP Logical Operator**

Welcome to PHP

Welcome to class 12

**PHP String Operators**

PHP has two operators that are specially designed for strings.


Operator	Name	Example	Result
.	Concatenation	\$txt1 . \$txt2	Concatenation of \$txt1 and \$txt2
.=	Concatenation assignment	\$txt1 .= \$txt2	Appends \$txt2 to \$txt1

**Example:**

```

<html>
  <head>
    <title>String operator</title>
  </head>
  <body>
    <b> PHP String Operator</b> <br/>
    <?php
      $fName="Devendra";
      $LName=" Chapagain";
      echo "Your First Name: $fName <br/>";
      echo "Your Last Name: $LName <br/>";
      $fName.=$LName;
      echo "Your Full Name: $fName <br/>";
    ?>
  </body>
</html>

```

**Result:**

**PHP String Operator**

Your First Name: Devendra

Your Last Name: Chapagain

## PHP Array Operators

The PHP array operators are used to compare arrays.

Operator	Name	Example	Result
+	Union	$\$x + \$y$	Union of $\$x$ and $\$y$
==	Equality	$\$x == \$y$	Returns true if $\$x$ and $\$y$ have the same key/value pairs
===	Identity	$\$x === \$y$	Returns true if $\$x$ and $\$y$ have the same key/value pairs in the same order and of the same types
!=	Inequality	$\$x != \$y$	Returns true if $\$x$ is not equal to $\$y$
<>	Inequality	$\$x <> \$y$	Returns true if $\$x$ is not equal to $\$y$
!==	Non-identity	$\$x !== \$y$	Returns true if $\$x$ is not identical to $\$y$

### Example:

```
<html>
  <head>
    <title>Array operator</title>
  </head>
  <body>
    <b> PHP Array Operator</b> <br/>
    <?php
      $arr1 = array(1,2,3);
      $arr2 = array(2,4,6,8);
      print_r($arr1); //print_r function displays
      print_r($arr2); // the information about variables
      if($arr1!=$arr2)
        echo "<br/>Both array are not same <br/>";

    ?>
  </body>
</html>
```

### Result:



#### PHP Array Operator

```
Array ( [0] => 1 [1] => 2 [2] => 3 ) Array ( [0] => 2 [1] => 4 [2] => 6 [3] => 8 )
```

```
Both array are not same
```

## PHP Conditional Operators

The PHP conditional operators are used to return a value depending on conditions. Syntax to declare PHP conditional operator is given below:

```
Result = condition ? expression1 : expression2;
```

Where,

The value of result is expression1 if condition is true

The value of result is expression2 if condition is false

### Example:

```
<html>
  <head>
    <title>Conditional operator</title>
  </head>
  <body>
    <b> PHP Conditional Operator</b> <br/>
    <?php
      $x=5; $y=7;
      echo"x=$x y=$y ";
      $greater=($x>$y)?$x:$y;
      echo "<br/>Greate number is: $greater <br/>";
    ?>
  </body>
</html>
```

### Result:



### PHP Conditional Operator

x=5 y=7

Greate number is: 7

## PHP POST Method

This is the built in PHP super global array variable that is used to get values submitted via HTTP POST method. The array variable can be accessed from any script in the program; it has a global scope.

This method is ideal when you do not want to display the form post values in the URL.

A good example of using post method is when submitting login details to the server.

It has the following syntax.

```
<?php
  $_POST['variable_name'];
?>
```

HERE,

"\$\_POST[...]" is the PHP array

"variable\_name" is the URL variable name.

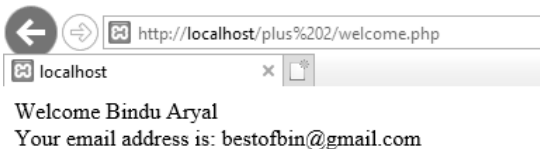
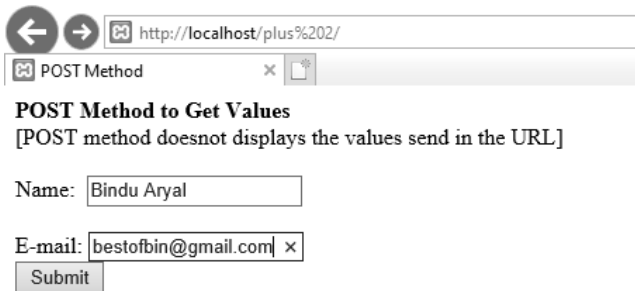
## Example to illustrate the concept of POST method

File: Home.php

```
<html>
  <head>
    <title>Post Method </title>
  </head>
  <body>
    <b> Post Method to Get Values</b><br>
    [POST method doesnot displays the values send in the URL]
    <form action="welcome.php" method="post"> <br>
    Name: &nbsp;<input type="text" name="name"><br><br>
    E-mail: <input type="text" name="email"><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

File: Welcome.php

```
<html>
  <body>
    Welcome <?php echo $_POST["name"]; ?><br>
    Your email address is: <?php echo $_POST["email"]; ?>
  </body>
</html>
```



## PHP GET Method

This is the built in PHP super global array variable that is used to get values submitted via HTTP GET method.

The array variable can be accessed from any script in the program; it has a global scope.

This method displays the form values in the URL. It's ideal for search engine forms as it allows the users to book mark the results.

It has the following syntax.

```
<?php
$_GET['variable_name'];
?>
```

HERE,

“\$\_GET[...]” is the PHP array

“variable\_name” is the URL variable name.

### Example to illustrate the concept of GET method

File: Home.php

```
<html>
  <head>
    <title>Get Method </title>
  </head>
  <body>
    <b> Get Method to Get Values</b><br>
    [GET method displays the values send in the URL]
    <form action="welcome.php" method="GET"> <br>
    Name: &nbsp;<input type="text" name="name"><br><br>
    E-mail: <input type="text" name="email"><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

File: Welcome.php

```
<html>
<body>
Welcome <?php echo $_GET["name"]; ?><br>
Your email address is: <?php echo $_GET["email"]; ?>
</body>
</html>
```

## Result

Get Method to Get Values  
[GET method displays the values send in the URL]

Name:

E-mail:

Welcome Dev  
Your email address is: devcpgn@gmail.com

## Difference between GET and POST Method

S.N.	GET Method	POST Method
1	Only limited amount of data can be sent because data is sent in header.	Large amount of data can be sent because data is sent in body.
2	GET request is not secured because data is exposed in URL bar.	POST request is secured because data is not exposed in URL bar.
3	GET request can be bookmarked.	POST request cannot be bookmarked
4	GET is essentially used for fetching the information	The purpose of POST method is to update the data.
5	It can be cached	It cannot be cached
6	GET method is the default method if the method is not specified in the form.	POST method must be specified in the form. It is not default method.

## Some Solved Problems

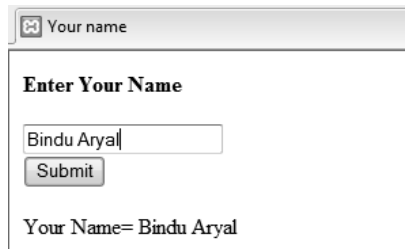
1. Write a PHP code to enter your name and display it.

```
<html>
  <head>
    <title>Your name </title>
  </head>
  <body>
    <b>Enter Your Name</b> <br/>
    <form method="POST">
```

```

<input type="text" name="name" /><br/>
<input type="submit" value="Submit"/>
</form>
<?php
    $name=$_POST['name'];
    echo "Your Name= $name";
?>
</body>
</html>

```

**Result:**


Your name

**Enter Your Name**

Bindu Aryal

Submit

Your Name= Bindu Aryal

**2. Write a PHP code to display the factorial of a number given by user.**

```

<html>
    <head>
        <title>Factorial </title>
    </head>
    <body>
        <b>Factorial Calculation Program</b> <br/>
        <form name="Dev" method="POST">
            Input a Number:
            <input type="Number" name="number" /><br/>
            <input type="submit" value="Calculate"/>
        </form>
        <?php
            $n=$_POST['number'];
            $fact=1;
            for($i=1;$i<=$n;$i++)
            {
                $fact=$fact*$i;
            }
            echo "The factorial of $n = $fact";
        ?>
    </body>
</html>

```

**Result:**

Factorial

**Factorial Calculation Program**

Input a Number:

The factorial of 6 = 720

**3. Write a PHP code to display all even numbers up to 50**

```
<html>
  <head>
    <title>even numbers </title>
  </head>
  <body>
    <b>Displaying Even Numbers Up to 50</b> <br/>
    <hr>
    <?php
      $a=2;
      for($i=1;$i<=25;$i++)
      {
        echo"$a ,";
        $a=$a+2;
      }
    ?>
  </body>
</html>
```

**Result:**

even numbers

**Displaying Even Numbers Up to 50**

---

2 ,4 ,6 ,8 ,10 ,12 ,14 ,16 ,18 ,20 ,22 ,24 ,26 ,28 ,30 ,32 ,34 ,36 ,38 ,40 ,42 ,44 ,46 ,48 ,50 ,

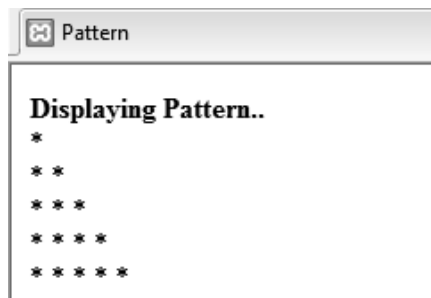
**4. Write a PHP code to display the following patterns**

```
*
* *
* * *
* * * *
* * * * *
<html>
```

```

<head>
<title>Pattern </title>
</head>
<body>
<b>Displaying Pattern.. </b> <br/>
<?php
    for ($x=1;$x<=5;$x++)
    {
        for ($y=1;$y<=$x;$y++)
        {
            echo "* ";
        }
        echo "<br>";
    }
?>
</body>
</html>

```

**Result:****5. Write a PHP code to display the multiplication table of a number**

```

<html>
<head>
<title>Multiplication Table </title>
</head>
<body>
<b>Multimpication Table from 1 to 5 </b> <br/>
<table border="1" cellpadding="3" cellspacing="0">
<?php
    for ($i=1;$i<=5;$i++)
    {
        echo "<tr>";
        for ($j=1;$j<=10;$j++)

```

```

    {
    echo "<td>$i * $j = ".$i*$j."</td>";
    }
    echo "</tr>";
    }
?>
</table>
</body>
</html>

```

**Result:****Multiplication Table from 1 to 5**

1 * 1 = 1	1 * 2 = 2	1 * 3 = 3	1 * 4 = 4	1 * 5 = 5	1 * 6 = 6	1 * 7 = 7	1 * 8 = 8	1 * 9 = 9	1 * 10 = 10
2 * 1 = 2	2 * 2 = 4	2 * 3 = 6	2 * 4 = 8	2 * 5 = 10	2 * 6 = 12	2 * 7 = 14	2 * 8 = 16	2 * 9 = 18	2 * 10 = 20
3 * 1 = 3	3 * 2 = 6	3 * 3 = 9	3 * 4 = 12	3 * 5 = 15	3 * 6 = 18	3 * 7 = 21	3 * 8 = 24	3 * 9 = 27	3 * 10 = 30
4 * 1 = 4	4 * 2 = 8	4 * 3 = 12	4 * 4 = 16	4 * 5 = 20	4 * 6 = 24	4 * 7 = 28	4 * 8 = 32	4 * 9 = 36	4 * 10 = 40
5 * 1 = 5	5 * 2 = 10	5 * 3 = 15	5 * 4 = 20	5 * 5 = 25	5 * 6 = 30	5 * 7 = 35	5 * 8 = 40	5 * 9 = 45	5 * 10 = 50

**6. Write a PHP code to display the simple interest.**

```

<html>
  <head>
    <title>Simple interest </title>
  </head>
  <body>
    <b>Simple Interest Calculator</b> <br/>
    <table border="1">
      <tr><td>
        <form name="SI" method="POST">
          Input Principle:
          <input type="Number" name="p" /><br/>
          Input Rate:
          <input type="Number" name="r" /><br/>
          Input Time:
          <input type="Number" name="t" /><br/>
          <input type="submit" value="Calculate SI"/>
        </form>
      <?php
        $p=$_POST['p'];
        $t=$_POST['t'];
        $r=$_POST['r'];

        $si=($p*$t*$r)/100;

```

```

        echo "Simple Interest= $si";
        ?>
    </td></tr>
</table>
</body>
</html>

```

**Result:**

**Simple Interest Calculator**

Input Princiaple: 1200

Input Rate: 8

Input Time: 4

Calculate SI

Simple Interest= 384

**Exercise**

1. What is variable? How variables are declared in PHP? Explain with example.
2. Write the rules for naming variable in PHP.
3. Write short notes on echo and print statements.
4. Describe the data types used in PHP.
5. What is operator? Describe PHP operators with examples.
6. Describe PHP increment and decrement operator with example.

**Programming Exercise**

1. Write a PHP code to enter any two values and display their sum, difference and product.
2. Write a PHP code to enter length and breadth of a rectangle and display area and perimeter.
3. Create a registration form including Name, Address, Phone NO, email address, Qualification, Gender and Country and get the values entered by user using POST method.

## Database Connectivity

The primary purpose of database is to store information. Using database, you can retrieve data, insert data, update data and delete existing data. There are many database programs available, such as, Microsoft Access, Microsoft SQL server, Oracle, FoxPro, DB2, MySQL and many others. Almost all databases use tables to store information. A table is a two dimensional matrix that is used to store information. Database can consist of several tables. You can use database of your choice to store your information.

## Connecting Server Side Script to Database

Before a client and server can exchange data and information, they must first establish a connection. PHP is best with MySQL, so we use MySQL database for the connection. PHP is built by web developers and supported by large community. MySQL is a powerful database management system. Combination of both technologies has a strong banding together. The combination of PHP and MySQL server has the following strength:

- Low cost
- Ease of use
- Cross platform compatibility
- Stability
- Strong user communities
- Fast development
- Can be embedded with HTML

## Steps Involved in Making Connection with MySQL Database

Installing XAMPP server on your computer requires the following steps. The software packet contains the web server Apache, the relational database management system MySQL and the scripting language PHP. XAMPP server can be installed and used with a single executable file quickly and easily, functioning as a local test system for Linux, Windows, and Mac OS.

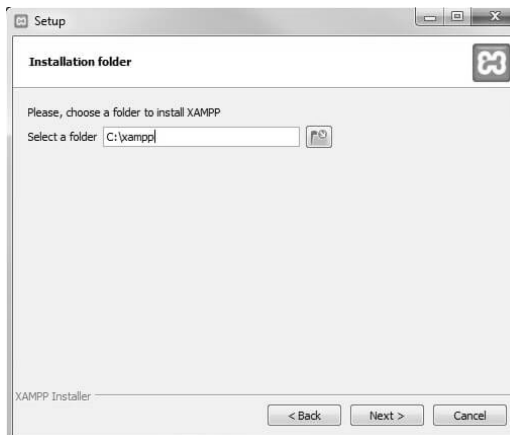
**Step 1:** Download XAMPP server for windows/ Linux from internet. Versions with PHP 5.5, 5.6, or 7 are available for download.

**Step 2:** Once you download software, Run .exe file.

**Step 3:** Start up setup wizard. Click on 'Next' to configure the installation settings.



**Step 4:** Choose the installation directory for XAMPP server to be installed. Default folder c:\xampp will be created. Click Next



**Step 5:** Start the installation process by clicking to start installation. This process can take several minutes in total.



**Step 6:** Your Firewall may interrupt the installation process to block the some components of the XAMPP. Use the corresponding check box to enable communication between the Apache server and your private network or work network.

**Step 7:** Complete installation. Once all the components are unpacked and installed, you can close the setup wizard by clicking on 'Finish'. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.



## The XAMPP Control Panel

Controls for the individual components of your test server can be reached through the XAMPP Control Panel. It allows you to start or stop individual modules. The XAMPP Control Panel also offers you various other buttons, including:

**Config:** allows you to configure the XAMPP as well as the individual components

**Netstat:** shows all running processes on the local computer

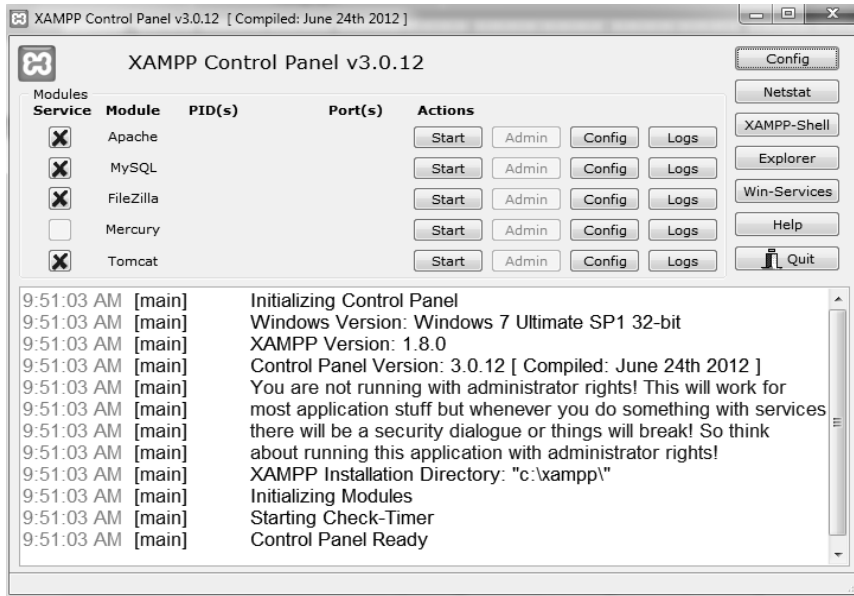
**Shell:** opens a UNIX shell

**Explorer:** opens the XAMPP folder in Windows Explorer

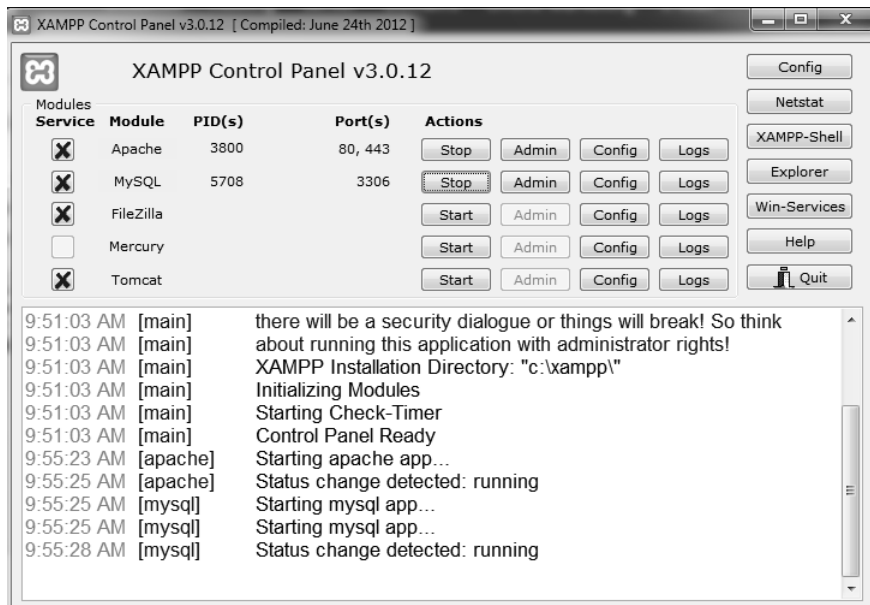
**Services:** shows all services currently running in the background

**Help:** offers links to user forums

**Quit:** closes the XAMPP Control Panel



**Starting Modules:** Individual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under 'Actions'.



## Testing XAMPP Installation

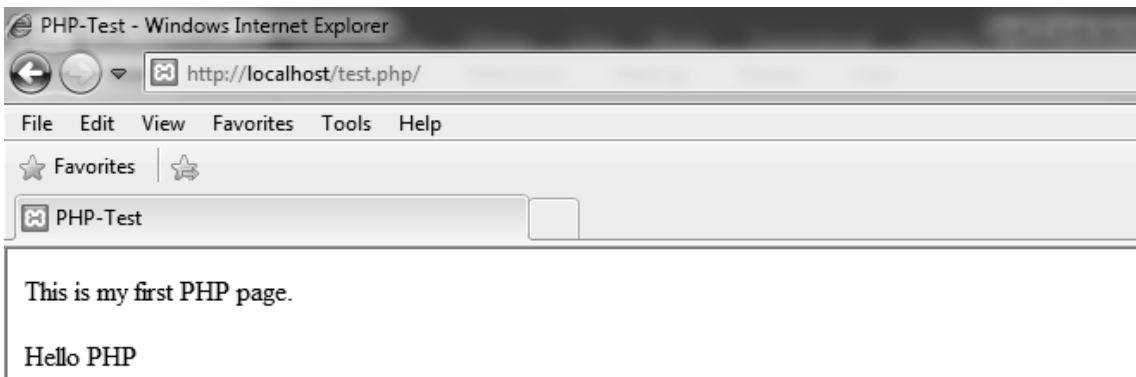
To check whether your test server is installed and configured correctly, you have the option to create a PHP test page, store them on your XAMPP's local host, and retrieve them via the web browser.

Open the XAMPP directory through the 'Explorer' button in the Control Panel and choose the folder htdocs (C:\xampp\htdocs for standard installations). This directory will store file data collected for web pages that you test on your XAMPP server. The htdocs folder should already contain data to help configuration of the web server.

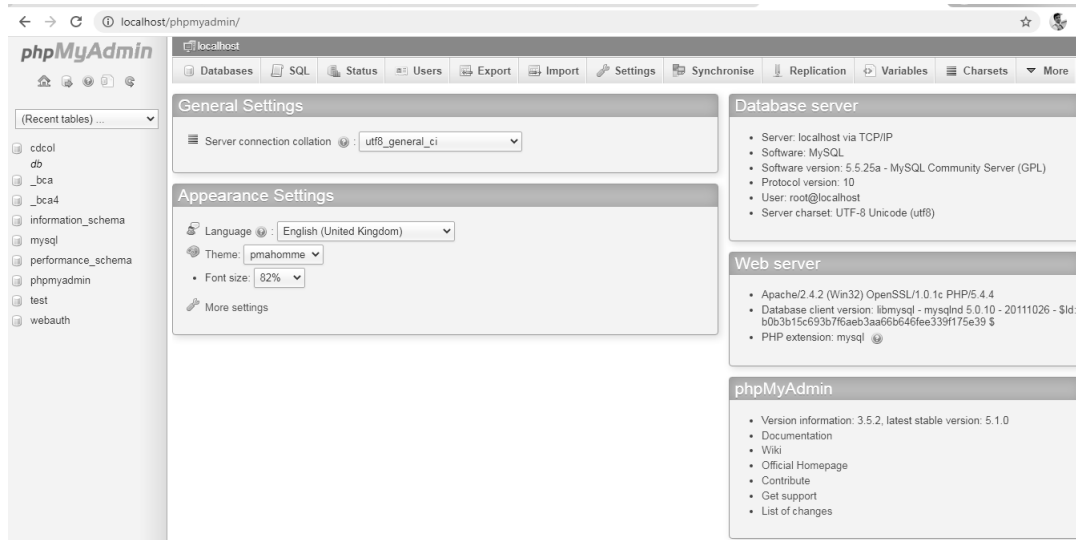
You can create a new PHP page easily by using the following content in your favorite editor and storing it as test.php in your 'test' folder (C:\xampp\htdocs\test.php):

```
<html>
  <head>
    <title>PHP-Test</title>
  </head>
  <body>
    This is my first PHP page.<br/>
    <?php
      echo '<p>Hello PHP</p>';
    ?>
  </body>
</html>
```

The last step now is to open your web browser and load your PHP page via localhost/test/test.php. If your browser window displays the words 'Hello PHP', then you've successfully installed and configured your XAMPP.



To check whether MySQL server running or not type: <http://localhost/phpmyadmin/> in your browser. It will display the following:



Your MySQL server is ready. Now you can run SQL queries and establish database connection with proper server side scripting.

## Making SQL Queries

SQL (Structure Query Language) is the database language. SQL is used to perform operations on the records stored in the database such as updating records, deleting records, creating and modifying tables, views, etc. SQL is just a query language; it is not a database. To perform SQL queries you must install any database like: MySQL, FoxPro, PostGre SQL, SQL Server, Oracle, DB2, etc. SQL queries are used to:

- Create new database and tables
- Retrieve data from database
- Insert records in database
- Update records in database
- Delete records from database

### MySQL Create Database

To create a new database you use the CREATE DATABASE statement.

#### Syntax:

```
CREATE DATABASE Database_Name;
```

#### Example:

```
CREATE DATABASE dbClass12;
```

## MySQL Create Table

To create a new table within a database, you use the MySQL CREATE TABLE statement.

### Syntax:

```
CREATE TABLE table_name (column_name column_type);
```

### Example:

```
create TABLE tb_students(  
RollNO integer(10),  
Name varchar(20),  
Address varchar(20),  
PhoneNo varchar(20));
```

## MySQL Insert

### Syntax:

```
INSERT INTO table_name (field1, field2,...fieldN )  
VALUES (value1, value2,...valueN );
```

To insert string data types, it is required to keep all the values into double or single quotes. For example "value".

### Example:

```
INSERT INTO `tb_students`(`RollNO`, `Name`, `Address`, `PhoneNo`) VALUES (1, "Aayan  
Chapagain", "Kawasoti", '9816400360');
```

## MySQL Select

### Syntax:

```
SELECT field1, field2,...fieldN  
FROM table_name1, table_name2...  
[WHERE Clause]  
[LIMIT N]
```

- You can use one or more tables separated by comma to include various conditions using a WHERE clause, but the WHERE clause is an optional part of the SELECT command.
- You can fetch one or more fields in a single SELECT command.
- You can specify star (\*) in place of fields. In this case, SELECT will return all the fields.
- You can specify any condition using the WHERE clause.
- You can limit the number of returns using the LIMIT attribute.

**Example:**

```
SELECT * FROM `tb_students`
```

+ Options		RollNO	Name	Address	PhoneNo
<input type="checkbox"/>	Edit  Copy  Delete	1	Aayan Chapagain	Kawasoti	9816400360
<input type="checkbox"/>	Edit  Copy  Delete	2	amyog Bhusal	Gaindakot	9847413133

**MySQL Where Clause**

We have used the SQL SELECT command to fetch data from a MySQL table. We can use a conditional clause called the WHERE Clause to filter out the results. Using this WHERE clause, we can specify a selection criteria to select the required records from a table.

**Syntax:**

```
SELECT field1, field2,...fieldN table_name1, table_name2...
```

```
[WHERE condition1 [AND [OR]] condition2.....
```

**MySQL Delete**

If you want to delete a record from any MySQL table, then you can use the SQL command DELETE FROM.

**Syntax:**

```
DELETE FROM table_name [WHERE Clause]
```

- If the WHERE clause is not specified, then all the records will be deleted from the given MySQL table.
- You can specify any condition using the WHERE clause.
- You can delete records in a single table at a time.

**MySQL Update**

There may be a requirement where the existing data in a MySQL table needs to be modified. You can do so by using the SQL UPDATE command.

**Syntax:**

```
UPDATE table_name SET field1 = new-value1, field2 = new-value2
```

```
[WHERE Clause]
```

- You can update one or more field altogether.
- You can specify any condition using the WHERE clause.
- You can update the values in a single table at a time.

## Creating SQL Database with Server Side Scripting

### PHP `mysqli_connect()` Function

The `mysqli_connect()` function opens a new connection to the MySQL server. MySQLi is an extension that only supports MySQL databases. It allows access to new functionalities found in MySQL systems (versions 4.1. and above), providing both an object-oriented and a procedural interface.

#### Syntax:

`mysqli_connect (host, username, password, dbname, port, socket);`

Parameter	Description
Host	Specifies a host name or an IP address
Username	Specifies the MySQL username
password	Specifies the MySQL password
dbname	Specifies the default database to be used
Port	Optional. Specifies the port number to attempt to connect to the MySQL server
socket	Optional. Specifies the socket or named pipe to be used

#### Example:

```
<?php
function OpenCon()
{
    $dbhost = "localhost";
    $dbuser = "root";
    $dbpass = "";
    $db = "dbclass12";
    $conn = new mysqli($dbhost, $dbuser, $dbpass,$db)
    or die("Connect failed: ". $conn -> error);

    return $conn;
}

function CloseCon($conn)
{
    $conn -> close();
}
?>
```

#### Explanation:

- `mysqli_connect()`. It establishes the connection with the specified database.
- If the connection failed, it would give the message Connection failed. The die function will print the message and then exit out of the script.
- If the connection is successful, it will display "Connected successfully."
- `mysqli_close` function closes the database connection

Example of User Login (User login must match the value from MySQL database)

First create table "users" in a database. And insert some values users table. Second create a login form using html

### Login.html

```
<html>
<head>
<title>login page</title>
</head>
<body>
<form name="login" action="login.php" method="POST">
<table align="center" bgcolor="#bfff00">
<tr>
<td> Login page</td>
</tr>
<tr>
<td> User Name:</td>
<td><input type="text" name="username" placeholder="User Name "></td>
</tr>
<tr>
<td>Password:</td>
<td><input type="password" name="pwd"placeholder="password"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Login">
<input type="reset" value="cancle"></td>
</tr>
</table>
</body>
</html>
```



### Login.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "demo";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error)
{
    die("Connection failed: " . $conn->connect_error);
}
if(isset($_POST['username']))
```

```

{
    $user=$_POST['username'];
    $pass=$_POST['pwd'];

    $query = "SELECT * FROM users where username='".$user.'" and
password='".$pass.'" ";
    $result = mysqli_query($conn,$query);
    if ( mysqli_num_rows($result)>0)
    {
        echo "Login successfull ";
        header('Location: welcome.php');//redirects to the welcome.php
page
    }
    else
    {
        echo"User Name or Password Not Matched";
    }
}
?>

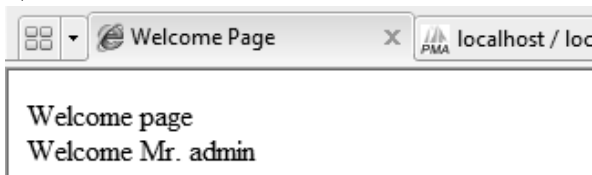
```

### Welcome.php

```

<html>
<body>
Welcome page
<?php
$name= $_GET["name"];
echo"Welcome Mr. $name ";
?>
</body>
</html>

```



## Fetching Data Sets Getting Data About Data

Data can be fetched from MySQL tables by executing SQL SELECT statement through PHP function `mysql_query`. You have several options to fetch data from MySQL. The most frequently used option is to use function `mysql_fetch_array()`. This function returns row as an associative array, a numeric array, or both. This function returns FALSE if there are no more rows. The below code illustrates how to fetch the data sets from database tables.

## Displaying Queries in Tables

You can display the table rows from database to your web page in tabular format.

Create a table in database and insert the values

+ Options

		Name	Class	RollNo	Marks	PhoneNo
<input type="checkbox"/>	Edit  Copy  Delete	Aayan	XI	1	78.5	9816400360
<input type="checkbox"/>	Edit  Copy  Delete	Samyog	XI	2	73.5	9847413133
<input type="checkbox"/>	Edit  Copy  Delete	Grisma	XII	3	80	9842365215

Check All /  Uncheck All With selected:

Write a HTML and PHP code to fetch the query.

```

<html>
  <title>Displaying queries in tables</title>
<body>
  Displaying queries in tables <br>

  <?php
  $servername = "localhost";
  $username = "root";
  $password = "";
  $dbname = "db_bca";
  $conn = new mysqli($servername, $username, $password, $dbname);
  if ($conn->connect_error)
  {
    die("Connection failed: " . $conn->connect_error);
  }
  $sql = "SELECT * FROM student";
  if($result = mysqli_query($conn, $sql)){
    if(mysqli_num_rows($result) > 0){
      echo "<table border=1>";
      echo "<tr>";
      echo "<th>RollNo</th>";
      echo "<th>Name</th>";
      echo "<th>Class</th>";
      echo "<th>Marks</th>";
      echo "<th>PhoneNo</th>";
      echo "</tr>";
      while($row = mysqli_fetch_array($result)){
        echo "<tr>";
        echo "<td>" . $row['RollNo'] . "</td>";
        echo "<td>" . $row['Name'] . "</td>";
        echo "<td>" . $row['Class'] . "</td>";
        echo "<td>" . $row['Marks'] . "</td>";
        echo "<td>" . $row['PhoneNo'] . "</td>";
        echo "</tr>";
      }
      echo "</table>";
      // Free result set
      mysqli_free_result($result);
    }
  }
  }
  
```

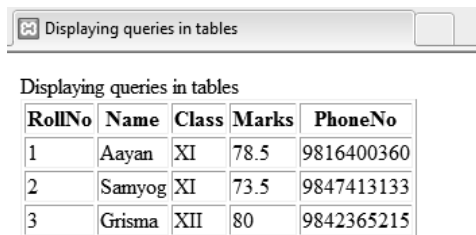
```

    } else{
        echo "No records matching your query were found.";
    }
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($conn)
;
}

?>
</body>
</html>

```

### Result:



The screenshot shows a web browser window with the title "Displaying queries in tables". The browser displays a table with the following data:

RollNo	Name	Class	Marks	PhoneNo
1	Aayan	XI	78.5	9816400360
2	Samyog	XI	73.5	9847413133
3	Grisma	XII	80	9842365215

### Points to Remember

- Scripting language is used to create dynamic web pages.
- Popular scripting languages used today are: PHP, JavaScript, python, Perl, ruby, jQuery.
- Server side scripting is used to create connection to the database.
- Client side scripts runs on client browser.
- Server side scripts runs on serve machine.
- Web pages can be static or dynamic.
- JavaScript is used to make web pages interactive. It is a client side scripting language that runs on client browser.
- Scripts cannot run individually, it requires other programming language where it can be embedded.
- JavaScript variable can hold a value of any data type. JavaScript is loosely typed language.
- Users can input data using HTML forms.
- The users data can be validated in both client side and server side
- Query is a small JavaScript library which helps to add animation and effects easily
- PHP (Hypertext Preprocessor) is a widely-used, open source scripting language
- PHP files have a file extension of ".php"
- SQL (Structure Query Language) is the database language.

## ✓ Multiple Choice Questions

- Dynamic web page ....
  - is same every time whenever it displays
  - generates on demand by a program or a request from browser
  - both (a) and (b)
  - none of above
- What will be the output of following JavaScript code
 

```
int a==2;
int b=4;
int ans=a+b;
print(ans);
```

  - 2
  - 6
  - 0
  - error
- Among the following, which one is a ternary operator?
  - +
  - :
  - 
  - ?:
- What will be the output of the following JavaScript code?
 

```
int a=0;
for(a;a<5;a++);
console.log(a);
```

  - 0
  - error
  - 4
  - 5
- Which one of the following is the correct way for calling the JavaScript code?
  - Preprocessor
  - Triggering Event
  - RMI
  - Function/Method
- Which one of the following symbol is used for creating comments in the javascript:
  - \\
  - //
  - \\* \\*
  - \\* \*/
- Which one of the following keywords is used for defining the function in the JavaScript?
  - Void
  - init
  - Main
  - Function
- How do you focus a particular part of the HTML page in JavaScript?
  - hover()
  - focus()
  - on()
  - focuson()
- Which of the following can't be done with client-side JavaScript?
  - Validating a form
  - Sending a form's contents by email
  - Storing the form's contents to a database file on the server
  - None of the above
- What is the correct JavaScript syntax to write "Hello World"?
  - System.out.println("Hello World")
  - println ("Hello World")
  - document.write("Hello World")
  - response.write("Hello World")
- Inside which HTML element do we put the JavaScript?
  - <js>
  - <scripting>
  - <script>
  - <javascript>
- Which types of image maps can be used with JavaScript?
  - Server-side image maps
  - Client-side image maps
  - Server-side image maps and Client-side image maps
  - None of the above
- Which of the following event fires when the form element loses the focus:
  - onfocus
  - onblur
  - onclick
  - ondblclick
- JavaScript is interpreted by \_\_\_\_\_
  - Client
  - Server
  - Object
  - None of the above
- What will be the output of following code:
 

```
<script type="text/javascript">
x=4+"4";
document.write(x);
</script>
```

  - 44
  - 8
  - 4
  - Error output
- Choose the correct option.
  - HTML form elements are used for taking user input.
  - HTML form elements are defined inside <form> tag.
  - HTML form elements can be of different types.
  - All of these.

17. Which of the following tag is used for drop down list?
- <select>
  - <text>
  - <textarea>
  - <dropdown>
18. Correct HTML Tag For The Largest Heading Is
- <h6>
  - <heading>
  - <h1>
  - <head>
19. What Tag Is Used To Display A Picture In A HTML Page?
- Picture
  - Img
  - Src
  - Image
20. PHP Stands for?
- Hypertext Processor
  - Hyper Markup Processor
  - Hyper Markup Preprocessor
  - Hypertext Preprocessor
21. PHP is an example of \_\_\_\_\_ scripting language.
- Server-side
  - Client-side
  - Browser-side
  - In-side
22. PHP scripts are enclosed within \_\_\_\_\_
- <php> . . . </php>
  - <?php . . . ?>
  - ?php . . . ?php
  - <p> . . . </p>
23. Which of the following method sends input to a script via a URL?
- Get
  - Post
  - Both
  - None
24. PHP files have a default file extension of.
- .html
  - .xml
  - .php
  - ph
25. Which of the looping statements is/are supported by PHP?
- for loop
  - while loop
  - do-while loop
  - foreach loop
- (i) and (ii)
  - (i), (ii) and (iii)
  - All of the mentioned
  - None of the mentioned
26. Which of the following php statement is correct to store the value of num variable
- int \$num = 15;
  - int mum = 15;
  - \$num = 15;
  - 111 = \$num;
- Both (i) and (ii)
  - All of the mentioned.
  - Only (iii)
  - Only (i)
27. What will be the output of the following php code
- ```
<?php
$num = 1;
$num1 = 2;
print $num . "+" . $num1 ;
?>
```
- 3
  - 1+2
  - 1.+2
  - Error
28. When you use the \$\_GET variable to collect data, the data is visible to.....
- none
  - only you
  - everyone
  - selected few
29. Which one of the following should not be used while sending passwords or other sensitive information?
- GET
  - POST
  - REQUEST
  - NEXT
30. Which one of the following databases has PHP supported almost since the beginning?
- Oracle Database
  - SQL
  - SQL+
  - MySQL
31. Which one of the following statements is used to create a table?
- CREATE TABLE table\_name (column\_name column\_type);
  - CREATE table\_name (column\_type column\_name);
  - CREATE table\_name (column\_name column\_type);
  - CREATE TABLE table\_name (column\_type column\_name);
32. Which one of the following statements instantiates the mysqli class?
- mysqli = new mysqli()
  - \$mysqli = new mysqli()
  - \$mysqli->new(mysqli())
  - mysqli->new(mysqli())
33. Which one of the following statements can be used to select the database?
- \$mysqli=select\_db('databasename');
  - mysqli=select\_db('databasename');
  - mysqli->select\_db('databasename');
  - \$mysqli->select\_db('databasename');
34. Which one of the following methods is responsible for sending the query to the database?
- query()
  - send\_query()
  - sendquery()
  - query\_send()

35. Which one of the following method is used to retrieve the number of rows affected by an INSERT, UPDATE, or DELETE query?
- num\_rows()
  - affected\_rows()
  - changed\_rows()
  - new\_rows()
36. The ..... statement is used to delete a table.
- DROP TABLE
  - DELETE TABLE
  - DEL TABLE
  - REMOVE TABLE
37. What is the full form of SQL?
- Structured Query List
  - Structure Query Language
  - Sample Query Language
  - None of these
38. Which of the following is not a DDL command?
- TRUNCATE
  - ALTER
  - CREATE
  - UPDATE
39. How many Primary keys can have in a table?
- Only 1
  - Only 2
  - Depends on no of Columns
  - Depends on DBA
40. Which command is used to change the definition of a table in SQL?
- CREATE
  - UPDATE
  - ALTER
  - SELECT

#### Answer Key

|      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| 1.b  | 2.d  | 3.d  | 4.d  | 5.d  | 6.b  | 7.d  | 8.b  | 9.c  | 10.c |
| 11.c | 12.b | 13.b | 14.a | 15.a | 16.d | 17.a | 18.c | 19.b | 20.d |
| 21.a | 22.b | 23.a | 24.c | 25.c | 26.c | 27.b | 28.c | 29.a | 30.d |
| 31.a | 32.b | 33.d | 34.a | 35.b | 36.a | 37.b | 38.d | 39.a | 40.c |

## Exercise

### ■ Short Answer Questions

- What is SQL? Why it is used?
- Differentiate between GET and POST method
- Explain any five SQL queries with example.
- Explain PHP mysqli\_connect() function.
- Why database connection is required? Explain the database connection process in PHP and MySQL.
- Write a SQL query for creating a database named inventory. Include the following tables within the inventory database.
  - SalesMan(salsmanID, Name, City, Commission)
  - Oders (OrderNo, PurchaseAmt, OrderDate, CustomerID, SalesManID)
  - Customer (CustomerID, CustomerName, City, SalesManID)
- Write a SQL Query to insert at least 5 values for each above created tables.
- Write a SQL Query to create a bank database
  - Create a table named Bank Detail with the following columns:
    - Bank Name
    - Location
    - Contact\_No
    - Bank\_Type
    - Bank\_Info

- b. Insert any two values in above table
  - c. Display all the information of bank using SELECT statement
  - d. Delete one record from the table “Bank Detail”
9. Write a SQL statement to create tables in database in the given format.

**Table: Employee**

| Column Name   | Data type |
|---------------|-----------|
| EmployeeID    | Number    |
| FirstName     | Varchar2  |
| LastName      | Varchar2  |
| Email         | Varchar2  |
| PhoneNo       | Varchar2  |
| Salary        | Number    |
| Date_Of_Join  | Date      |
| Department_ID | Number    |

10. Insert the given values in above table

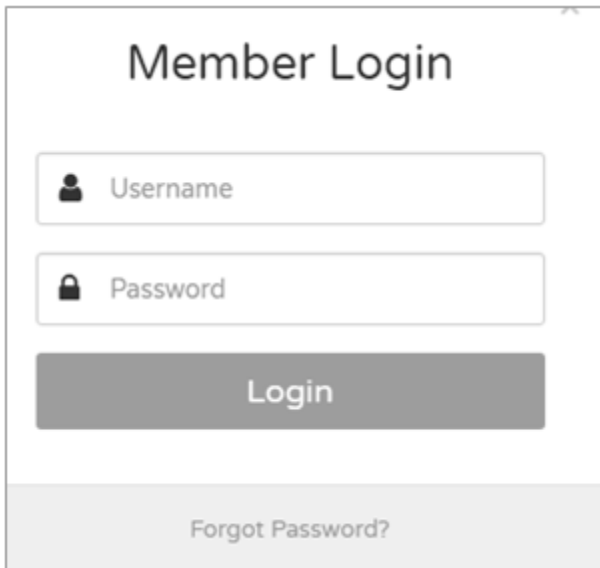
| EmployeeID | FirstName | LastName  | Email            | PhoneNo     | Salary | Date_Of_Join | Dept_Id |
|------------|-----------|-----------|------------------|-------------|--------|--------------|---------|
| 500        | Dipendra  | Silwal    | dip@gmail.com    | 9863254521  | 85000  | 2070/01/05   | 10      |
| 501        | Deepak    | Adhikari  | deepak@gmail.com | 9825626524  | 87000  | 2072/05/12   | 12      |
| 502        | Hari      | Thapaliya | hari@gmail.com   | 98548257895 | 95000  | 2065/01/14   | 15      |
| 503        | Shanti    | Sharma    | shanti@gmail.com | 9865326587  | 75000  | 2071/06/05   | 15      |
| 504        | Tej       | Aryal     | tez@gmail.com    | 9855025632  | 83000  | 2074/04/12   | 10      |

From above table run the following Query

- a. Display the last name, email and phone number of all the employees
- b. Display the first name, last name phone number and salary of employees who are from department number 15
- c. Display all the information of employee whose salary is greater than 50000
- d. Change the department number of Shanti to 12
- e. Change the salary of employees who belong to department number 10 to 90000
- f. Remove the information of employee who are earning below 50000

## Project Work

1. Create a registration form including: name, address, phone number, gender, academic qualification, country, email address, submit and cancel button. Use appropriate HTML tags and add style wherever required.
2. Design a login form as below and write PHP code to login. Use appropriate CSS and JavaScript wherever necessary.



Member Login

Username

Password

Login

Forgot Password?

3. Write a code in PHP to find the sum of  $3 \times 2$  matrices





# Programming in C



## LEARNING OBJECTIVES

After going through this unit, you will be able :

- ◆ to know more about c programming behind basic concepts
- ◆ to use library and user defined functions in a program
- ◆ to learn the concept and advantages of using function
- ◆ to use storage variables in a program
- ◆ to understand the recursive function and use of recursion
- ◆ to Differentiate between structure and union
- ◆ to use pointers concept in a program
- ◆ to learn to perform various FILE input/output operations

## Review of C Programming Concept

You have already studied about basic concept of C programming in class XI. C language is a very good language to introduce yourself to the programming world, as it is a simple procedural language. Initially C language was mainly used for writing system level programs, like designing operating systems, but there are other applications as well which can be very well designed and developed using C language, like Text Editors, Compilers, Network Drivers, database, operating system, utilities software etc. In previous book of class XI you have studied the following topics. Let us discuss in short.

### Programming Concept

- C Programming Languages
- Introduction and features of C Language
- Structure of C program
- C Preprocessor and Header Files
- Character Set used in C
- Use of Comments
- Identifiers, Keywords and Tokens
- Basic Data Types in C
- Constants and Variables
- Type of Specifier
- Simple and Compound Statements
- Operators and Expressions: arithmetic, relational, logical, assignment, unary and conditional operators
- Input/output (I/O) Functions
- Selection Control Statement: Decisions (if, if-else, if-else-if, nested and, switch)
- Iteration Control Statement: Looping (while, do while, for nested)
- Array: definition, types (1D and 2D), matrix addition and subtraction
- String: definition and string function: strlen(), strcat(), strcmp(), strcmp(), strcpy(), strlen(),strupr()

### C Programming Languages

C is a general-purpose programming language, and is used for writing programs. Program is a set of instructions to perform specific task. It is a high level programming language and easy to learn. It consists of just 32 keywords. It is possible to design operating system, graphical applications, drivers, compilers etc. Almost most of the programming languages like Java, C++, C#, etc. follow C programming syntax. C programming language provides the core concepts (like loop, array, string, function, structure, file handling, etc.) for beginners.

## Features of C Language

You are familiar with most concepts of the C language in the class XI. If you haven't learnt some basics than you must be go through that before learning to some extra concepts of C programming language. C is widely used language. It provides many features such as:

- Simple to learn and understand
- Simple in size only having 32 keywords
- Portable or machine independent
- Provides a lot of inbuilt functions
- Faster execution
- Extensible
- Reusable

## Structure of C program

The following program is the first program presented in the textbook of class XI. For detail description refer to class XI book.

```
#include<stdio.h>
#include<conio.h>
main( )
{
int x=5; //variable decleration
printf(" value of x=%d,"x);
}
```

## C Preprocessor and Header Files

Preprocessor directives are the grammar of C. It is used for handling source file and macro definitions in the program. Header file is a file that is defined to be included at the beginning of a program in C language that contains the definitions of data types and declarations of variables used by the functions in the program is called header file. Some of the header files and associated functions in that file are discussed later in the chapter function.

## Character Set used in C

The character set of C programming consists of following elements:

Alphabet: A to Z and a to z.

Number: 0 to 9.

Special Characters: ; : { , " ' |  
 \ } > < / ~ \_ [ ]  
 ! \$ ? \* # ^ @ &

## Use of Comments

In C program, we can declare comments in two ways:

1. **Single Line Comment:** If we want to place only one line of comment somewhere in the program, we can write comment after double forward slashes (//). This is just one line comment hence it is called Single Line Comment. Single line of comment does not have ending slashes. For example:

```
//This is just testing.
```

2. **Block Comment/ Multi Line Comment:** If we want to place some blocks of comments in a program, we can write comment starting by forward slash asterisk (/\*) and write multiple lines of comment and ending with asterisk forward slash (\*). This is used for multiple lines of comment as a single block hence it is called Block Comment. This comment has starting as well as ending portion. For example:

```
/* This is just for testing.  
   It takes two values,  
   and generate greatest number.  
*/
```

## Identifiers, Keywords and Tokens

The basic element recognized by the compiler is known as tokens. A token is source-program text that the compiler does not break down into component elements. The keywords like int, float, if, for etc., identifiers like main, printf, void etc., constants like a, b, c etc., string literals like name, address, phone etc., and operators like &&, |, ! etc. are the common examples which is created by using one or more tokens.

## Basic Data Types in C

There are two types of data types in C. They can be classified as:

1. **Primary Data Types**
  - Char
  - Int
  - Float
  - Double
  - void
2. **Secondary Data types**
  - Array
  - Pointer
  - Structure
  - Union
  - Enum

## Constants and Variables

In a program you have to deal with so many values. Values can be either constant or variable. A constant is a fixed value which cannot be changed during the program execution. The value of variable can be changed during program execution. For example 10 and 5 are constants. If you define  $x=10$  and  $y=5$  then  $x$  and  $y$  are variables. Variables can hold a value at a time.

## Type of Specifier

The input and output data are formatted by specific pattern. These patterns are generated by using specific tokens in C programs. These tokens used to format data are called specifier. Some of the mostly used specifiers are:

1. Escape Sequence
2. Format Specifier

## Statements

A smallest executable entity within a program code is called a statement. An instruction or one line of code written to do a specific task in a program is called programming statement. Statements are the basic building blocks of C programming language. These statements enable the computer to carry out some calculation or perform logical comparison between values and variables. A programming statement may consist of keyword, constant, variables, operators, control statement, data type, library function, user-defined function etc.

## Operators and Expressions

An operator is a sign or symbol, which performs an operation or evaluation on one or more operands. Operands are values or variables declare within program. For example,  $5+10$  where '+' sign is an operator and 5 and 10 are operands and '+' sign indicates addition operation between the operands. The different types of operators used in C program are as follows:

- Arithmetic operators
- Relational operators
- Logical operators
- Assignment operators
- Conditional operators
- Bitwise operators
- Comma operator
- Increment/Decrement Operator
- Ternary Operator

## Input/output (I/O) Functions

C program also can take input and produce output. For input and output management, C provides the following Input/Output functions.

The Input/Output functions of C programming can be categorized into following types:

- Formatted I/O functions
- Unformatted I/O functions

Some of the formatted I/O function are as follows:

| S.N. | Function | Description                                                                                                                                                                                 |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | printf() | It is an output function that prints a character or string or numeric values on the screen. It offers %d, %c, %s, %f and others for integer decimal, character, string and float data type. |
| 2.   | scanf()  | It is an input function that reads the input from keyboard and different data can be entered like int, float, char, string etc. into C program.                                             |

Some of the Unformatted I/O functions are as follows:

| S.N. | Function  | Description                                                                                                                                                                                                                              |
|------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | getch()   | It is an input function which reads only one character at a time i.e. one keystroke without echoing input character on the screen. Without echoing means, it does not display the character which you press from keyboard on the screen. |
| 2.   | getche()  | It is an input function which reads only one character at a time i.e. one keystroke with echoing input character on the screen. With echoing means, it displays the character which you press from keyboard on screen.                   |
| 3.   | putch()   | It is an output function that outputs a single character on the screen.                                                                                                                                                                  |
| 4.   | getchar() | It is an input function to read a character from keyboard.                                                                                                                                                                               |
| 5.   | putchar() | It is an output function to print a character on the monitor.                                                                                                                                                                            |
| 6.   | gets()    | It is an input function that reads a single or multiple words from keyboard.                                                                                                                                                             |
| 7.   | puts()    | It is an output function that prints a single or multiple words on monitor.                                                                                                                                                              |

## Selection Control Statement: Decisions (if, if-else, if-else-if, nested and, switch)

These control statements determine the flow of control in a program and enable us to specify the order in which the various instructions in a program are to be executed by computer. The control structures are the very core part of programming language. Normally, high level procedural programming languages require three basic control statements:

1. Sequential Control Structure
2. Decisions Control Structure
  - If
  - If else
  - Nested if else
  - If else ladder
  - switch

### 3. Looping Control Structure

- For loop
- While loop
- Do while loop

#### Array:

An array is a collective name given to a group of similar quantities. . An array can be defined as number of memory locations, each of which can store the same data type and which can be references through the same variable name. There are two type of array declaration.

- One-Dimensional Array
- Two-Dimensional Array

#### String:

C string can be defined as the one-dimensional array of characters terminated by a null character ('\0'). There are different string manipulation functions. These are inbuilt functions defined within <string.h> header file. Some of the most used string functions are as follows:

- Strlen
- Strupr
- Strlwr
- Strcat
- Strcmp
- Strev
- Strcpy

Now let us move to further new concepts of C programming that you have not learnt in class XI.

## Functions

A function is a block of code that performs a specific task. Functions are independent block of statements that performs the definite task when called. Every C program can have many functions. If there is only one function in a program that must be main () function. It is because the execution of a program starts from the main () function. In any program in C there is only one main function in which other functions are called. Function helps to divide the large program into small blocks which helps to write the large and complex program easily. The function contains the set of programming statements enclosed by {}.

### The Main Function

The main() function is the entry point of any C program. It is the point at which execution of program is started. When a C program is executed, the execution control goes directly to the main() function. Every C program have a main() function.

**Syntax of main function:**

```
void main()  
{  
    .....  
    .....  
}
```

**void:** is a keyword, void means nothing, whenever we use void as a function return type then that function return nothing.

In place of void we can also use int return type of main() function, at that time main() return integer type value.

**main:** is a name of function which is predefined function in C library.

**Advantages of using Functions**

The advantages of using functions are given below:

- Avoid repetition of codes.
- Increases program readability by avoiding rewriting of same code at multiple locations
- Divide a complex problem into simpler ones.
- Chance of error is reduced.
- Modifying a program becomes easier by using function.
- Length of program is reduced.
- Program development and debugging is faster.
- Function keeps the code organized, easy to read and make it reusable
- It saves time and effort of the programmer
- Functions enable us to hide the implementation details of a program, e. g., we have used library functions such as strlen, strcpy, sqrt etc. without ever knowing how they are implemented.

**Concept of Library and User Defined Functions**

There are two types of function in c, they are:

- Library Function/ Built-in function
- User Defined Function

**1. Library Function/Built-in Function/Predefined Function**

Library functions are that type of function which are already defined, compiled and stored in header file of c library. We can easily use these functions by including their header file (files with .h extensions are called header files such as stdio.h) in our program. Library functions makes programs more compact, easier to write and easier to maintain. It is not necessary to declare and define this function. You can just call the function.

**Syntax to call the function:**

```
Function_Name (Parameter1, Parameter2 ...);
```

**For example:**

```
Strlen(string);
```

C language has several libraries that include predefined functions to make programming easier. In C language, header files contain the set of predefined standard library functions. You can request to use a header files in your program by including it with the C preprocessing directive “#include”. The entire header file has a ‘.h’ an extension. In C, all the header files must necessarily end with the “.h” extension. By including a header file, you can use its contents in your program. Some of the predefined functions in C library are:

**Functions defined in math.h header file**

The math.h header defines various mathematical functions. All the functions available in this library take double as an argument and return double as the result.

| Function            | Description                                              | Example                                             |
|---------------------|----------------------------------------------------------|-----------------------------------------------------|
| abs(num)            | Returns the absolute value of num                        | int a=-10;<br>printf(" abs value of a= %d",abs(a)); |
| ceil(num);          | Round up the given number.                               | int y=3.3;<br>printf("y=%d",ceil(y));               |
| floor(num);         | Rounds down the given number.                            | printf("\n%f",floor(3.6));                          |
| sqrt(num);          | Returns the square root of given number.                 | printf("\n%f",sqrt(9));                             |
| pow(base, exponent) | Returns the power of given number.                       | printf("%f",pow(2,3));                              |
| log(num)            | Returns the natural logarithm (base-e logarithm) of num. | x = 2.8;<br>printf("log(%f) = %f", x, log(x));      |

**Functions Defined in stdio.h header file**

The header file stdio.h stands for Standard Input Output. It has the information related to input/output functions. Some of the functions in stdio.h are listed below:

| Function | Description                                                                   | Example                   |
|----------|-------------------------------------------------------------------------------|---------------------------|
| printf() | It is used to print the strings, integer, character etc on the output screen. | printf("Hello");          |
| scanf()  | It reads the character, string, integer etc from the keyboard.                | scanf("%d",&x);           |
| fopen()  | It opens the file for read/write operation                                    | fp=fopen("book.txt","w"); |
| fclose() | It closes the opened file after read/write operation is over                  | fclose(fp);               |
| getc()   | It reads a character from a file                                              | ch=getc(fp);              |
| putc()   | It writes a character to the file                                             | putc(char, fp);           |

### Functions Defined in string.h header file

The functions for string manipulation are defined in the string.h header file. Some of the functions in string.h are:

| Function                   | Description                                                                         | Example                                                           |
|----------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| strlen(string)             | Returns the length of string                                                        | int len=strlen("Hello World");                                    |
| strcpy(destination,source) | Copies the contents of source string to destination string.                         | strcpy(s1,s2);                                                    |
| strupr(string)             | Returns the given string character in uppercase                                     | strupr("computer");                                               |
| strlwr(string)             | Returns the given string character in lowercase                                     | strlwr("COMPUTER");                                               |
| strrev(string)             | Returns the reverses string                                                         | strrev("Note Book");                                              |
| strcmp(string1, string2)   | Compares the first string with second string                                        | strcmp(s1,s2);                                                    |
| strcat(string)             | Joins the first string with second string.                                          | strcat(s1,s2);                                                    |
| strchr(str, ch)            | Returns a pointer to the first occurrence of ch in str, or NULL if ch is not found. | char s1[20]="Computer Science";<br>printf("a=%s",strchr(s1,'a')); |

### Functions Defined in stdlib.h header file

The header file stdlib.h stands for Standard Library. It has the information of memory allocation/freeing functions. Some of the functions defined in the stdlib.h are listed below:

| Function | Description                                              | Example                                                                                                              |
|----------|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| rand()   | It returns a random integer value                        | #include "stdio.h"                                                                                                   |
| malloc() | It allocates the memory during execution of program      | #include <stdlib.h>                                                                                                  |
| free()   | It frees the allocated memory.                           | int main()                                                                                                           |
| abort()  | It terminates the C program.                             | {                                                                                                                    |
| exit()   | It terminates the program and does not return any value. | printf("Random value 1: %d\n",<br>rand());<br>abort();<br>printf("Random value 2 : %d",<br>rand());<br>exit(0);<br>} |

### Functions Defined in Time.h header file

The time.h header file contains various functions for manipulating date and time. Some of the functions defined in time.h header files are listed below:

| Function                                  | Description                                                               | Example                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| clock_t clock(void)                       | Returns the number of clock ticks elapsed since the program was launched. | <pre>#include &lt;stdio.h&gt; #include &lt;time.h&gt; int main () {     clock_t start_t, end_t;     int i;     start_t = clock();     printf("Starting of loop, start_t =     %ld\n", start_t);     for(i=0; i&lt; 100000000; i++) {     }     end_t = clock();     printf("End of loop, end_t = %ld\n",     end_t); }</pre> |
| char *ctime(const time_t *timer)          | Returns a string representing the local time based on the argument timer. | <pre>#include &lt;stdio.h&gt; #include &lt;time.h&gt; int main () {     time_t curtime;     time(&amp;curtime);     printf("Current time = %s",     ctime(&amp;curtime)); }</pre>                                                                                                                                            |
| struct tm *localtime(const time_t *time); | converts calendar time into local time                                    | <pre>#include &lt;stdio.h&gt; #include &lt;time.h&gt; int main () {     time_t time1;     struct tm *info;     time( &amp;time1 );     info = localtime( &amp;time1 );     printf("Current local time and     date: %s", asctime(info)); }</pre>                                                                             |

### Functions defined in ctype.h header file

The ctype.h header file of the C Standard Library declares several functions that are useful for testing and mapping characters.

| Function  | Syntax         | Description                                   |
|-----------|----------------|-----------------------------------------------|
| isalpha() | isalpha(int c) | Checks if the passed character is alphabetic. |

**Example:**

```
#include <stdio.h>
#include <ctype.h>
int main ()
{
    int x = '1';
    if(isalpha(x))
        printf("%c is an alphabet\n", x);
    else
        printf("%c is not an alphabet\n", x);
}
```

| Function  | Syntax         | Description                                                  |
|-----------|----------------|--------------------------------------------------------------|
| isdigit() | isdigit(int c) | Checks if the passed character is a decimal digit character. |

**Example:**

```
#include <stdio.h>
#include <ctype.h>
int main ()
{
    int x = 'a';
    int y = '7';
    if(isalpha(x))
        printf("%c is an alphabet\n", x);
    else
        printf("%c is not an alphabet\n", x);
    if(isdigit(y))
        printf("%c is a digit\n", y);
    else
        printf("%c is not a digit\n", y);
}
```

| Function  | Syntax             | Description                                                |
|-----------|--------------------|------------------------------------------------------------|
| islower() | int islower(int c) | Checks whether the passed character is a lowercase letter. |

**Example:**

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
int main ()
{
    int ch = 'A';
    if( islower(ch) ) {
        printf(" %c is lowercase character\n", ch);
    } else {
        printf(" %c is not lowercase character\n", ch);
    }
    getch();
}
```

| Function  | Syntax             | Description                                                |
|-----------|--------------------|------------------------------------------------------------|
| isupper() | int isupper(int c) | Checks whether the passed character is a uppercase letter. |

**Example:**

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
int main ()
{
    int ch = 'A';
    if( isupper(ch) ) {
        printf(" %c is uppercase character\n", ch);
    } else {
        printf(" %c is not uppercase character\n", ch);
    }
    getch();
}
```

| Function  | Syntax             | Description                                             |
|-----------|--------------------|---------------------------------------------------------|
| isprint() | int isprint(int c) | Returns TRUE if 'c' is a printable character else FALSE |

**Example:**

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
int main ()
```

```

{
  int x = 'a';
  int y = '\n';
  if(isprint(x))
    printf("%c can be printed\n", x);
  else
    printf("%c can't be printed\n", x);
  getch();
}

```

| Function   | Syntax              | Description                                                  |
|------------|---------------------|--------------------------------------------------------------|
| isxdigit() | int isxdigit(int c) | Checks if the passed character is a decimal digit character. |

Example:

```

#include <stdio.h>
#include <conio.h>
#include <ctype.h>
int main ()
{
  int n = '5';
  if( isdigit(n) )
    printf("%c is a digit\n", n);
  else
    printf("%c is not a digit\n", n);
  getch();
}

```

## Advantages of Library Functions

The advantages of library functions are given below:

- Easy to use and they are 100% accurate as they are tested multiple times
- Each library function performs specific operation
- Reduction in program size
- Library functions optimizes the performance of programs
- Using library functions saves development time because the code is already written in header files.

## User Defined Function

The functions that we create in a program are known as user defined functions or in other words you can say that a function created by user is known as user defined function. The user defined function requires the following components:

- Function prototype (Declaration of function)
- Function Body (Definition of function)
- Function call
- Return statement (optional)

#### Syntax to create user defined function:

```
return_type function_name (argument list)
{
    Function body
}
```

#### Example:

```
int sum(int a, int b)
{
    int sum=a+b;
    return sum;
}
```

The user defined function has following characteristics:

- Function has unique name given by programmer.
- It performs a specific task
- A function can perform its task without interference with other parts of the program. That is function is independent.
- Function can receive the value from the calling program (caller).
- Calling program can pass values to function
- Function may return the value to the calling program.
- The code written within a function is not executed until the function is called.
- When it is called, the program can send information to the function in the form of one or more arguments. Argument is a program data needed by the function to perform its task.
- Once function is written, it can be called as many times as needed.

#### Advantages of User Defined Functions

The advantages of user defined functions are:

- Code can be reused multiple times
- Complex program can be divided into different sub programs.
- Easy to debug and maintain
- It makes easy to understand the logic of the program
- As user defined function can be called many times, this avoids the writing of same code again and again.

## Difference between Library Function and User Defined Functions

| Basis of Comparison  | User Defined function                                                            | Library Function                                                      |
|----------------------|----------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Function creation    | These functions are created by user as per their own requirement.                | These functions are not created by user as their own.                 |
| Storage              | User-defined functions are not stored in library file.                           | Library Functions are stored in special library file.                 |
| Renaming a function  | The name of function can be changed at any time                                  | The name of a function cannot be changed                              |
| Function Declaration | User must declare and define these functions                                     | Declaration and definition of function is not required                |
| Function definition  | User defined function is not a predefined function.                              | Library function is predefined function in a header file              |
| Function call        | User defined function are part of the program which is called at compile runtime | Library functions are part of header file which is called at runtime. |
| Example              | sum(), factorial(), check(), area(), perimeter(), etc.                           | printf(), scanf(), strlen(), sqrt(),strupr(), clrscr(), etc.          |

## Function Definition, Prototype, Call and Return Statements

### Function Definition

A function definition provides the actual body of the function. Function definition contains the block of code to perform a specific task. When a function is called, the control of the program is transferred to the function definition and the compiler starts executing the codes inside the body of a function. This contains all the statements to be executed.

#### Syntax:

```
returnType functionName(type1 argument1, type2 argument2, ...)
{
    //body of the function
}
```

### Function Prototype/ Function Declaration

Function declaration tells the compiler about a function's name, return type, and parameters. It does not contain function body. A function prototype gives information to the compiler that the function may later be used in the program. Before using the function in a program it must be declared. This process of declaring a function is called function prototype. The function declaration indicates how many functions are used within a program.

#### Syntax:

```
returnType functionName(type1 argument1, type2 argument2, ...);
```

#### Example:

```
int sum(int a, int b);
```

Function declaration is not required if the function call is after the function body. Function declaration is required if the function call is before the function definition.

For example:

| Function call before the function body                                                                                                                                                                                    | Function call after the function body                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>#include&lt;stdio.h&gt; void abc(int); // declaration int main() {     int a=5;     abc(a); //function call before body of function } void abc(int a ) // function body {     printf("The value of a=%d",a); }</pre> | <pre>#include&lt;stdio.h&gt; void abc(int a ) // function body {     printf("The value of a=%d",a); } int main() {     int a=5;     abc(a); //function call after function body }</pre> |

## Function Call

Function call means calling a function whenever it is required in a program. Whenever we call a function, it performs an operation for which it was designed. This calls the actual function. We can call the function any number of times in a program. If you call a function that is not defined then it will give an error. To call a function simply writes a name of function and required parameter list.

### Syntax:

NameofFunction(parameter list);

Example:

abc(x,y);

## Return Statement

The return is used to return from a function. It causes execution to return to the point at which the call to the function was made. If the function is of void type then, return statement is not required. The function which does not have return type can display the values in their own function body.

### Syntax:

return expression;

- expression can be variable or constant
- if function does not return any value we can simply write :
 

```
return;
```
- It returns a value
- Return type can be **void** if function is not returning any value
- Return type can be **int** if it returns integer value
- Return type can be **char** if it returns character value

## Types of User Defined Function

A user defined function, depending on whether arguments are present or not and whether a value is returned or not, may belong to one of the following categories.

- Functions with no arguments and no return values
- Functions with no arguments and with return values
- Functions with arguments and no return values
- Functions with arguments and with return values

### Functions with no Arguments and no Return Values

When a function has no arguments, it does not receive any data from the calling function. Similarly, when it does not return a value, the calling function does not receive any data from the called function.

Syntax:

```
void main()
{
function_Name();
....
}
void function_Name()
{
.....
}
```

### Example:

/\* Program to find the sum of two numbers using function with no argument and no return values \*/

```
#include <stdio.h>
#include <conio.h>
void add(); //function declaration
int main()
{
    add();
    getch(); //function call
}
void add() //function body
{
    int a=5,b=7;
    int c=a+b;
    printf("Total=%d\n",c);
}
```

## Functions with no Arguments and with Return Values

In this type of function, function has no arguments, it does not receive any data from the calling function, but it returns a value. It means one result will be sent back to the caller from the function.

Syntax:

```
void main()
{
function_Name();
....
}
void function_Name()
{
.....
Return statement
}
```

Example:

*/\* Program to find the sum of two numbers using Functions with no arguments and with return values \*/*

```
#include <stdio.h>
#include <conio.h>
int add(); //function declaration
int main()
{
    int sum;
        sum=add();
    printf("Sum=%d\n",sum);
    getch(); //function call
}
int add() //function body
{
    int x=5,y=7,c;
        c=x+y;
        return(c);
}
```

## Functions with arguments and no return values

Here function will accept data from the calling function as there are arguments, however, since there is no return type nothing will be returned to the calling program.

Syntax:

```

void main()
{
function_Name(arguments);
....
}
void function_Name(arguments)
{
.....
}

```

Example:

/\* Program to find the sum of two numbers using Functions with arguments and no return values \*/

```

#include <stdio.h>
#include <conio.h>
int add(int, int); //function declaration
int main()
{
    int x=5,y=7;
    add(x,y);
    getch(); //function call
}
int add(int a,int b) //function body
{
    int sum;
    sum=a+b;
    printf("Sum=%d",sum);
}

```

### Functions with arguments and with return values

Function with arguments and return value means both the calling function and called function will receive data from each other.

Syntax:

```

void main()
{
function_Name(arguments);
....
}
void function_Name(arguments)
{
Return statemetns

```

```
}
```

Example:

/\* Program to find the sum of two numbers using Functions with arguments and with return values \*/

```
#include <stdio.h>
#include <conio.h>
int add(int,int); //function declaration
int main()
{
    int sum;
        sum=add(5,7);
    printf("Sum=%d\n",sum);
    getch(); //function call
}
int add(int x,int y) //function body
{
    int c;
        c=x+y;
        return(c);
}
```

## Accessing a Function by Passing Values

### Function Arguments/Parameters

Function arguments or parameters are used to receive the required values by the function call. The number of argument must be matched with the called function and calling function. The function argument arguments are matched by positions of the arguments.

**Actual parameters:** The parameters that appear in a function calls.

**Formal parameters:** The parameters that appear in a function declaration.

### Parameter Passing Mechanism of Function:

1. Call by value (pass by value)
2. Call by reference (pass by reference)

#### 1. Call by Value (Pass by Value)

In this method the value of arguments are sent to the function. Any change made to the formal arguments does not change the actual arguments. If you want that the value of actual argument should not get changed in the function being called, pass the actual argument by value.

```
//program to calculate area using pass by value method
#include<stdio.h>
```

```

#include<conio.h>
int area(int,int);
int main()
{
int length, breadth, a;
printf("Enter length: ");
scanf("%d",&length);
printf("Enter breadth: ");
scanf("%d",&breadth);
a=area(length,breadth); // pass by value
printf("The area =%d",a);
getch();
}
int area(int x, int y) //function body
{
return (x*y);
}

```

**Output:**

```

Enter length: 7
Enter breadth: 4
The area =28

```

**2. Call by Reference (Pass by Reference)**

In this method the address of arguments are sent to the function. Any change made to the formal arguments will be reflected on actual arguments. If you want that the value of an actual argument should get changed the function being called, pass the actual argument by reference. If you want to return more than one value at a time, then use call by reference.

**//program to calculate area using pass by reference method**

```

#include<stdio.h>
#include<conio.h>
int area(int *x, int *y) //function body
{
int l,b;
l=*x;
b=*y;
return (l*b);
}
int main()
{
int length, breadth, a;
printf("Enter length: ");
scanf("%d",&length);
printf("Enter breadth: ");
scanf("%d",&breadth);
a=area(&length, &breadth); //Pass by reference
printf("The area =%d",a);
getch();
}

```

**Output:**

```
Enter length: 5
Enter breadth: 3
The area =15
```

**Difference between pass by value and pass by reference**

| Basis of Comparison | Pass by Value                                                           | Pass by Reference                                                              |
|---------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Return value        | Function cannot return more than one value at a time                    | Function can return more than one value at a time                              |
| Concept of pointer  | Concept of pointer cannot be used in call by value                      | Concept of pointer can be used in call by reference                            |
| Change in value     | Any changes made in formal arguments do not change the actual arguments | Any changes made in formal arguments will be reflected in the actual arguments |
| Value send          | A copy of data is send to the function                                  | The memory address of data is send to the function                             |

## Concept of Storage

In addition to data types, each variable has one more attribute known as storage class. Storage class specifies the lifetime and visibility of the variable. Lifetime is the length of time that variable holds a value. Visibility also called scope is the accessibility of that variable. The proper use of storage class makes program efficient and fast. These storage class variables are also called memory variables.

**Syntax:**

```
Storage_class datatype variableName;
```

**Example:**

```
extern int x;
```

There are four types of storage classes. They are:

| Storage Class | Purpose                                            |
|---------------|----------------------------------------------------|
| 1. auto       | It is default storage class                        |
| 2. external   | It is global variable                              |
| 3. static     | It is local variable                               |
| 4. register   | It is a variable which is stored inside a register |

### 1. auto

All variables declared inside a function without any storage class specifies are called automatic variables. All variables like int, char, float etc. are type of automatic variables. We can also use **auto** keyword to declare automatic variables. But this is generally not done.

- It is also called local/ internal variable
- Initially it takes garbage value by default
- It is initialized when program starts execution and destroy after completion of function.
- We can't access local variable outside the function.

```
//program to illustrate the concept
#include<stdio.h>
#include<conio.h>
void function1();
void function2();
int main()
{
    auto int a=40;
    function2();
    printf("a=%d\n", a);
    getch();
}
void function1()
{
    auto int a=50;
    printf("a=%d\n", a);
}
void function2()
{
    auto int a=60;
    printf("a=%d\n", a);
}
```

## 2. External

- External variable is also called global variable.
- It is declared above the main function.
- The initial value of external variable is zero.
- The keyword **extern** is specified in declaration but not in definition.
- Memory is allocated at the time of definition.

**//program to illustrate the concept**

```
#include<stdio.h>
#include<conio.h>
void function1();
int main()
{
    extern int k; //variable is accessible
    printf("k=%d", k);
    function1();
    getch();
}
int k=10;
void function1()
{
    k=k+10;
    printf("k=%d\n", k);
}
```

## 3. Static

Static is a local variable which is capable of returning a value even when control is transferred to the function call. The static variables have initial value zero and initialized only once on its lifetime.

**Example:** static int count = 10;

**//program to illustrate the concept**

```
#include<stdio.h>
#include<conio.h>
void next();
int main()
{
    int count;
    for(count=1;count<=5;count++)
    {
        next();
    }
}
void next( )
{
    static int i = 10; //initialized only once
    printf("i=%d \n", i);
    i++;
}
```

**Output:**

```
i=10
i=11
i=12
i=13
i=14
```

#### 4. Register

Register storage class can be applied only to local variable. The scope, lifetime and initial value of register variable are same as that of automatic variables. Register variables are stored in the CPU registers where as automatic variables are stored in memory. The variables stored in register can be accessed much faster than variables stored in memory. So if you required faster processing then assign the variable as register class.

**Syntax:**

```
register datatype variablename;
```

**Example:**

```
register int a;
```

**//program to understand the use of register variable**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    register int i;
    for(i=1;i<=200;i++)
    {
        printf("%d",i);
    }
    getch();
}
```

| Basis of Comparison | Auto                                                                                                                                                                                                                                                                       | External                                                                                                                                                                                                                                                                                                                                        | Static                                                                                                                                                                                                            | Register                                                                                                               |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| Definition          | Auto variables are always declared within a function in which they are declared. Hence they are also called local variables.                                                                                                                                               | A variable which can be accessed within a function and outside the main function. These variables are also named as Global variables.                                                                                                                                                                                                           | The variable static is constant and the value is continued in all the function call                                                                                                                               | These variable are stored in CPU register and hence they can be accessed faster than the one which is stored in memory |
| keyword             | auto                                                                                                                                                                                                                                                                       | extern                                                                                                                                                                                                                                                                                                                                          | static                                                                                                                                                                                                            | register                                                                                                               |
| declaration         | Inside the function                                                                                                                                                                                                                                                        | Outside of the main function                                                                                                                                                                                                                                                                                                                    | Inside the function                                                                                                                                                                                               | Inside the function                                                                                                    |
| Storage area        | Stack                                                                                                                                                                                                                                                                      | CPU memory                                                                                                                                                                                                                                                                                                                                      | CPU memory                                                                                                                                                                                                        | CPU register                                                                                                           |
| Initial value       | Garbage value(compiler assigns any value at compile time)                                                                                                                                                                                                                  | Zero                                                                                                                                                                                                                                                                                                                                            | Zero                                                                                                                                                                                                              | Garbage value                                                                                                          |
| Life time           | Up to the function only                                                                                                                                                                                                                                                    | Up to the entire program                                                                                                                                                                                                                                                                                                                        | The value of the variable persist between function calls                                                                                                                                                          | Up to that function only                                                                                               |
| Example             | auto int x; or<br>int x;                                                                                                                                                                                                                                                   | int x;<br>void main()<br>{<br>}<br>or<br>void main()<br>{<br>extern int x;<br>}                                                                                                                                                                                                                                                                 | static int x;                                                                                                                                                                                                     | register int x;                                                                                                        |
| Program             | <pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; void function1(); void function2(); int main() { int a=1000; function2(); printf("a=%d\n",a); getch(); } void function1() { int a=10; printf("a=%d\n",a); } void function2() { int a=100; printf("a=%d\n",a); }</pre> | <pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; int k; void function1(); void function2(); void function3(); int main() { k=20; function1(); function2(); function3(); getch(); } void function1() { k=k+10; printf("k=%d\n",k); } void function2() { k=k+20; printf("k=%d\n",k); } void function3() { k=k+30; printf("k=%d\n",k); }</pre> | <pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; void function1(); int main() { int i; for(i=0;i&lt;=5;i++) { function1(); } getch(); } void function1() { static int x=0; printf("x=%d\n",x); x=x+1; }</pre> | <pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; int main() { register int x; printf("X=%d", x); getch(); }</pre>  |

|        |                 |                      |                                        |                                    |
|--------|-----------------|----------------------|----------------------------------------|------------------------------------|
| output | a=100<br>a=1000 | k=30<br>k=50<br>k=80 | x=0<br>x=1<br>x=2<br>x=3<br>x=4<br>x=5 | The value of register variable x=5 |
|--------|-----------------|----------------------|----------------------------------------|------------------------------------|

## Concept of Recursion

Recursion is the process by which a function calls itself repeatedly until some specified condition has been satisfied. The process is used for repetitive computation in which each action is stated in terms of a previous result. The advantage of using recursive function is that it makes the code compact and elegant. It also simplifies the logic and hence makes the program easier to understand. However sometimes code written using recursive function is less efficient because recursion is a slow process as it involves many function calls. Not every problem can be solved using recursion. There exist non recursive solutions of most of the problems. To be recursive function the following condition should be satisfied:

- Function must be recursive in nature
- There must be stopping condition

### Example:

```
void abc()
{
    abc(); //function calls itself
}
```

### //program to find the factorial using recursive function

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int fact(int);
    int num;
    printf("Enter a number: ");
    scanf("%d",&num);
    printf("Factorial of %d is %d", num, fact(num));
    getch();
}
int fact(int n)
{
    if(n==0)
        return 1;
    else
        return (n*fact(n-1));
}
```

### Output:

```
Enter a number: 5
Factorial of 5 is 120
```

In above example, suppose, to find the factorial of 5.

Initially main() calls fact(5)

Since 5>0, fact(5) calls fact(4)

Since 4>0, fact(4) calls fact(3)

Since 3>0, fact(3) calls fact(2)

Since 2>0, fact(2) calls fact(1)

Since 1>0, fact(1) calls fact(0)

Since 0>0; FALSE; recursion stop and control returns to fact(1)

Now, every called function will return the value to the previous function

Recursive function put the values on stack which uses the concept of LIFO( Last In First Out)

/\* Program to display Fibonacci series using recursive function up to 10<sup>th</sup> term. For example:  
0,1,1,2,3,5,8..... \*/

```
#include<stdio.h>
#include<conio.h>
int fibonacci(int);
int main()
{
    int i;
    for (i = 0; i < 10; i++)
    {
        printf("%d, ", fibonacci(i));
    }
    getch();
}
int fibonacci(int i)
{
    if(i == 0)
        return 0;
    if(i == 1)
        return 1;
    return fibonacci(i-1) + fibonacci(i-2);
}
```

**Output:**

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

## Solved Problems

### 1. Write a program to enter a number and display its square using user defined function

```
//program to find the square of a given number
#include <stdio.h>
#include<conio.h>
int square(int);
int main()
{
    int num;
    int n;
    printf("Enter a number : ");
```

```

scanf("%d", &num);
n = square(num);
printf("The square of %d is : %d \n", num, n);
getch();
}
int square(int num)
{
return (num * num);
}

```

Output:

```

Enter a Number: 5
The square of 5 is: 25

```

## 2. Write a program to display the sum of natural numbers using recursive function

```

//program to display the sum of natural numbers using recursion
#include <stdio.h>
#include<conio.h>
int addNumbers(int n);
int main()
{
int num;
printf("Enter a Number: ");
scanf("%d", &num);
printf("Sum = %d", addNumbers(num));
getch();
}
int addNumbers(int n)
{
if (n != 0)
return (n + addNumbers(n - 1));
else
return n;
}

```

Output:

```

Enter a Number: 10
Sum = 55

```

## 3. Write a program to display whether a given number is positive or negative.

```

//Program to display whether the given number is positive or negative
#include <stdio.h>
#include<conio.h>
void check(int x)
{
if(x<0)
printf("The number is Negative");
else if(x>0)
printf("The Number is Positive");
else
printf("The Number is Neither Positive nor Negative");
}

```

```
int main()
{
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    check(num);
    getch();
}
```

**Output:**

```
Enter a number: -5
The Number is positive
```

**Exercise**

1. What is function? Write the advantages of using function.
2. Differentiate between user defined function and library function.
3. Differentiate between call by value and call by reference method.
4. What do you mean by recursive function? Explain with example.
5. What do you mean by storage class? Define auto, register and static storage class.
6. Define function. Explain the process of declaring and calling function.

**■ Programming Exercise**

1. Write a program to calculate the factorial of given number using function.
2. Write a program to calculate the factorial of given number using recursive function.
3. Write a program to enter length and breadth of a rectangle and display its area using function.
4. Write a program to find the sum of 'n' integer numbers using function.
5. Write a program to display the sum of n even numbers using function.
6. Write a program to display the Fibonacci series up to 10<sup>th</sup> term using recursive function.
7. Write a program to display series 2, 5, 8, 11, ..... up to n<sup>th</sup> term using function.
8. Write a program to enter a number and display whether a given number is odd or even using user defined function.
9. Write a program to display simple interest using user defined function.
10. Write a program to find whether a given number is positive or negative using function.
11. Write a program to enter a radius of a circle and display its area using user defined function.
12. Write a program to check a given number is even or odd using the function.
13. Write a program to check whether entered number is prime number or not using function.

## Structures and Unions

### Introduction to Structure

We may need to group different types of logically related data. For example, if we want to create a record of a person that contains name, age and height of that person then we can't use array because all the three data elements are of different types. Remember array can only store similar data types. Structures can be very useful in database management, for example: to maintain data about employees in an organization, books in library, items in a store etc. It makes very easy to insert, update, and delete data in database. It also makes the large programs efficient and systematic.

Structure is one which is used to store related fields of different data types. Simply structure is one that is capable of storing heterogeneous data. Data of different types can be grouped together under a single name using structures. The data element of a structure is referred to as members. In a nutshell-

- Structure is a user defined data types that stores different data types
- Each member must be unique
- No storage class can be used or attached with members
- We cannot initialize member variables within structures
- No memory is allocated for structure before declaring variable
- Structure is capable of storing heterogeneous data
- Structure are used to manage the database
- Structure makes the program systematic and efficient.

## Structure: Definition, Declaration, Initialization and Size of Structure

### Defining Structure

Definition of structure creates a template or formats that describe the characteristics of its members. All the variables that is declared will take the form of same template. The general syntax of a structure definition is:

```

struct structureName
{
    DataType Member 1;
    DataType Member 2;
    .....
    .....
    .....
    DataType Member N;
}variable1, variable2;
OR
struct structureName
{
    DataType Member 1;
    DataType Member 2;
    .....
    .....
    .....
    DataType Member N;
};
structureName variable1, variable2;
For example:
struct Student
{
    char name[30];
    int rollno;
    float marks;
};

```

## Declaring Structure Variable

By defining a structure we have only created a format, the actual use of structure will be when we declare variables based on this format. We can declare structure variables in two ways.

1. **Using structure definition:** In this we declare a variable while defining the structure template as-

```
struct Student
{
    char name[30];
    int rollno;
    float marks;
}stu1, stu2, stu3;
```

Here stu1, stu2 and stu3 are variables of type struct student

2. **Using structure tag:** We can also declare structure variable using structure tag. This can be written as-

```
struct Student
{
    char name[30];
    int rollno;
    float marks;
};
struct Student stu1, stu2, stu3;
```

## Initialization of Structure Variables

Initialization is the process of assigning first values to a variable. The initialization of structure variable is similar to that of arrays. All the values are given in curly braces and the number, order and the type of these values should be same as in the structure template definition. The initializing value can only be constant expression.

For example:

|                                                                                           |    |                                                                                                                             |    |                                                                                                                    |
|-------------------------------------------------------------------------------------------|----|-----------------------------------------------------------------------------------------------------------------------------|----|--------------------------------------------------------------------------------------------------------------------|
| <pre>struct Student { char name[30]; int rollno; float marks; }stu1={"Dev",1,74.5};</pre> | OR | <pre>struct Student { char name[30]; int rollno; float marks; }stu1; stu1.name="Dev"; stu1.rollno=1; stu1.marks=75.5;</pre> | OR | <pre>struct Student { char name[30]; int rollno; float marks; }stu1;  struct Student stu2={"Aayan", 2,72.5};</pre> |
|-------------------------------------------------------------------------------------------|----|-----------------------------------------------------------------------------------------------------------------------------|----|--------------------------------------------------------------------------------------------------------------------|

You cannot initialize members while defining the structure. For example

```
struct Student
{
    char name[30]="Bindu "; //Error
    int rollno=10; //Error; Invalid
    float marks=73.5; //Error
}stu1;
```

If the number of initializers is less than the number of members then remaining members are initialized with zero. For example:

```
struct Student
{
    char name[30];
    int rollno;
    float marks;
}stu1={"Aayan"};
```

Here, the member's rollno and marks of stu1 will be initialized to zero. This is equal to:

```
struct Student stu1={"Aayan", 0,0};
```

## Size of Structure

We may need to find the size of structure in some situations like reading or writing to files. To find out the size of structure you can use size of operator. You can either use the structure variable name or the tag name with the *struct* keyword.

For example:

```
struct Student
{
    char name[20]; //size 20 bytes
    int rollno;    // size 4 bytes
    float marks;  // size 4 bytes
}stu1;
```

Size of structure= 20+4+4=28 bytes (Sum of all data types)

//program to illustrate the size of structure

```
#include <stdio.h>
#include<conio.h>
struct Student
{
    char name[20];
    int rollno;
    float marks;
}stu1;
int main()
{
    int s;
    s=sizeof(stu1);
    //s=sizeof(struct Student);
    printf("The size of structure=%d",s);
}
```

**Output:**

```
The size of structure=28
```

**Note:** The size of int and float data type may vary in different compiler.

## Accessing Member of Structure

For accessing any member of structure variable, you can use dot (.) operator which is also known as the period or membership operator. Syntax for accessing structure member is:

```
Structure_variable.member;
```

**Example:**

```

struct Book
{
    char title[30];
    int price;
}book1;

```

Title of a book is accessed by- book1.title  
Price of the book is accessed by - book1.price

//program to show the concept of accessing the member of a structure.

```

#include <stdio.h>
#include<conio.h>
struct Book
{
    char title[30];
    int price;
}book1;

int main()
{
    printf("Enter the title of book:");
    scanf("%s",book1.title);
    printf("Ennter the price of book:");
    scanf("%d",&book1.price);

    printf("\n Title of book=%s ",book1.title);
    printf("\n Price of book=%d",book1.price);

    getch();
}

```

**Output:**

```

Enter the title of book:Computer
Ennter the price of book:365

Title of book=Computer
Price of book=365

```

## Array of Structure

We can declare array of structures where each element of array is of structure type. The array of structure is used to store large amount of similar records. For example, to store the record of 100 employees then the array of structure is used. Array of structure can be declared in two ways:

```

struct student
{
    char name[20];
    int roll;
    float marks;
}st[100];

struct student
{
    char name[20];
    int roll;
    float marks;
};

struct student st[100];

```

```

//Program to insert the records of 3 student and display
//them using structure
#include<stdio.h>
#include<conio.h>
#include<string.h>
struct student
{
    int RollNo;
    char Name[30];
    float marks;
}record[3];

int main()
{
    int i;
    //Inserting Records of Students
    for(i=0;i<3;i++)
    {
        printf("Record of Student %d\n ",i);
        printf("Enter the RollNo of Student:");
        scanf("%d",&record[i].RollNo);
        printf("Enter the Name of Student:");
        scanf("%s",record[i].Name);
        printf("Enter the marks of Student:");
        scanf("%f",&record[i].marks);
    }
    // Displaying Records of Students
    for(i=0; i<3; i++)
    {
        printf(" Records of STUDENT : %d \n", i);
        printf(" RollNO: %d \n", record[i].RollNo);
        printf(" Name: %s \n", record[i].Name);
        printf(" Percentage: %f\n\n",record[i].marks);
    }
    getch();
}

```

**Output:**

```

Record of Student 0
Enter the RollNo of Student:4
Enter the Name of Student:Kamal
Enter the marks of Student:45
Record of Student 1
Enter the RollNo of Student:5
Enter the Name of Student:Aayan
Enter the marks of Student:75
Record of Student 2
Enter the RollNo of Student:6
Enter the Name of Student:Grisma
Enter the marks of Student:74
Records of STUDENT : 0
RollNO: 4
Name: Kamal
Percentage: 45.000000

Records of STUDENT : 1
RollNO: 5
Name: Aayan
Percentage: 75.000000

Records of STUDENT : 2
RollNO: 6
Name: Grisma
Percentage: 74.000000

```

**Introduction to Union**

Union is a derived data type like structure and it can also contain members of different data types. The syntax used for definition of union, declaration of union variables and accessing members is similar to that used in structure, but the keyword 'union' is used instead of 'struct'.

**Union: Definition, Declaration**

Union is a derived data type like structure where member variables share a common memory space. In some situations in a program some variables may not be used. At that time union can share the memory occupied by that variable so instead of using extra memory, the variables share a common memory that are not used. The size of union is the largest data member which can hold the enough memory space.

The main difference between union and structure is in the way memory is allocated for the members. In structure each member has its own memory location where as members of union shares the same memory location. When a variable of type union is declared, compiler allocates sufficient memory to hold the largest data member. Since all members shares the same memory location, you can use only on member at a time.

**Advantages of Using Union**

- Union occupies less memory space compared to structure so you can save a memory a lot by using union.
- Union are very helpful for low level programming.

- Same memory can be used differently for different members of the union.
- The last variable can be directly accessed.
- Union can also be used for declaring array that can hold value of different data types.

### Syntax to declare union:

```
union nameofUnion
{
    Datatype member1;
    Datatype memebr2;
    Datatype memebr2;
}
```

### Example:

```
union u
{
    int a;
    float b;
    char c;
}
```

## Accessing Members of Union

You can access the union members using the same syntax used for structures. Union members can be accessed using dot (.) operator. Keep in mind that you can access only the recently used variable.

### //Program to access the union members

```
#include <stdio.h>
#include<conio.h>
union student
{
    int marks;
    char grade;
    float per;
}result;
int main()
{
    result.marks=87;
    printf("\n Marks =%d \n",result.marks);
    result.grade='A';
    printf("\n Grade =%c \n",result.grade);
    result.per=84.5;
    printf("\n Percentage =%f \n",result.per);
    getch();
}
```

### Output:

```
Marks =87
Grade =A
Percentage =84.500000
```

## Difference between Union and Structure

The difference between union and structure is given below:

| Basis of Comparison | Structure                                                                                                     | Union                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Keyword             | The keyword <b>struct</b> is used to define the structure                                                     | The keyword <b>union</b> is used to define the union                                                        |
| Internal storage    | The separate memory location is allotted to each member of the structure                                      | The same memory is allocated to all member of the structure.                                                |
| Size                | The total size of the structure is the sum of the size of every data member.                                  | The total size of the union is the size of the largest data member.                                         |
| Syntax              | <pre>struct struct_name{     type element1;     type element2;     .     . } variable1, variable2, ...;</pre> | <pre>union union_name{     type element1;     type element2;     .     . } variable1, variable2, ...;</pre> |
| Member access       | You can access any member at a time.                                                                          | You can access one member at a time                                                                         |
| Change in           | Changing the value of one data member will not affect other data members                                      | Changing the value of one data member will change the value of other data members.                          |
| Initialization      | All the members can be initialized at once.                                                                   | Only the first member can be initialized.                                                                   |
| Uses                | Structures are widely used in programming                                                                     | Unions are not used as much as structures in programming.                                                   |
| Example             | <pre>struct s { <br/><br/>};</pre>                                                                            | <pre>union u { <br/><br/>};</pre>                                                                           |

## Difference between Structure and Array

| Basis of Comparison  | Array                                                                                   | Structure                                                                                                                                       |
|----------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Defination           | Array is a collection of similar data.                                                  | Structure is a collection of dissimilar data                                                                                                    |
| Data accessed        | Array data is accessed using index                                                      | Structure elements are accessed using dot(.) operator                                                                                           |
| Memory Allocation    | Array allocates static memory                                                           | Structure allocates dynamic memory                                                                                                              |
| Speed                | It takes less time to access array elements                                             | It takes more time to access structure elements                                                                                                 |
| Hardware interaction | Arrays cannot be used in the program to interact with hardware                          | Structures can be used in the program to interact with hardware                                                                                 |
| Syntax               | <code>data_type arrayName[size];</code>                                                 | <pre>struct struct_name{     data_type element1;     data_type element2;     .     . } variable1, variable2, ...;</pre>                         |
| Example              | <pre>#include &lt;stdio.h&gt; #include&lt;conio.h&gt; int main() {     getch(); }</pre> | <pre>#include &lt;stdio.h&gt; #include&lt;conio.h&gt; struct person {     char Name[20];     float height; }; int main() {     getch(); }</pre> |

## Solved Problems

1. Write a program to enter ISBN No, Name, price and author of 5 books and display in proper format using structure.

```

/*program to enter ISBN No, Name, price and author of 5 books and display
in proper format using structure. */
#include<stdio.h>
#include<conio.h>
struct book
{
int isbn;
char book_title[50],author[50];
float price;
}b[5];
int main()
{
int i;
printf("Enter the details of book:\n");
for(i=0;i<5;i++)
{
printf("Enter Book %d detail ",i+1);
printf("\n -----");
printf("\n Enter ISBN No:");
scanf("%d",&b[i].isbn);
printf("Enter Book title:");
scanf("%s",b[i].book_title);
printf("Enter Author Name:");
scanf("%s",b[i].author);
printf("Enter price of book:");
scanf("%f",&b[i].price);
}
//Displaying results
printf("The book details are:");

for(i=0;i<5;i++)
{
printf("\n Book %d details:",i+1);
printf("\n ISBN : %d ",b[i].isbn);
printf("\n Book Title : %s ",b[i].book_title);
printf("\n Author : %d ",b[i].author);
printf("\n Price : %f ",b[i].price);
printf("\n -----");
}
getch();
}

```

Output:

```

C:\Users\97798\Desktop\Temp\structbook.e
Enter the details of book:
Enter Book 1 detail
-----
Enter ISBN No:12345
Enter Book title:Computer
Enter Author Name:Abcd
Enter price of book:450
Enter Book 2 detail
-----
Enter ISBN No:9857
Enter Book title:Accountancy
Enter Author Name:Xyz
Enter price of book:750
Enter Book 3 detail
-----
Enter ISBN No:5689
Enter Book title:Economics
Enter Author Name:Wxyz
Enter price of book:654
Enter Book 4 detail
-----
Enter ISBN No:1568
Enter Book title:Mathematics
Enter Author Name:Def
Enter price of book:842
Enter Book 5 detail
-----
Enter ISBN No:7895
Enter Book title:Marketing
Enter Author Name:David
Enter price of book:625
The book details are:
Book 1 details:
ISBN : 12345
Book Title : Computer
Author : 4223094
Price : 0
-----
Book 2 details:
ISBN : 9857
Book Title : Accountancy
Author : 4223202
Price : 0
-----

```

**2. Write a program to enter and print the roll no., name, age, address and marks of 15 students using structure.**


```

/*Program to enter and print the roll no., name, age, address and
marks of 15 students using structure.*/
#include<stdio.h>
#include<conio.h>
struct student
{
    int roll;
    char name[50],address[50];
    int age;
}stu[15];
int main()
{
    int i;
    // Enter student information
    for(i=0;i<15;i++)
    {
        printf("-----\n");
        printf("Enter Student %d information ",i+1);
        printf("\n Enter RollNo:");
        scanf("%d",&stu[i].roll);
        printf("Enter Name:");
        scanf("%s",stu[i].name);
        printf("Enter Address:");
        scanf("%s",stu[i].address);
        printf("Enter age:");
        scanf("%d",&stu[i].age);
    }
    //Displaying results
    printf("The Student information are:");

    for(i=0;i<15;i++)
    {
        printf("\n Student %d details:",i+1);
        printf("\n RollNo : %d ",stu[i].roll);
        printf("\n Name : %s ",stu[i].name);
        printf("\n Address : %s ",stu[i].address);
        printf("\n Age : %d ",stu[i].age);
        printf("\n -----");
    }
    getch();
}

```

## Output:

 Select C:\Users\97798\Desktop\Temp\strstu.exe

```
Enter Student 1 information
  Enter RollNo:10
Enter Name:Aayan
Enter Address:Kawasoti
Enter age:16
-----
Enter Student 2 information
  Enter RollNo:11
Enter Name:Grisma
Enter Address:chitwan
Enter age:15
-----
Enter Student 3 information
  Enter RollNo:12
Enter Name:Ankit
Enter Address:Butwal
Enter age:17
-----
Enter Student 4 information
  Enter RollNo:13
Enter Name:Bishal
Enter Address:pokhara
Enter age:18
The Student information are:
Student 1 details:
RollNo : 10
Name : Aayan
Address : Kawasoti
Age : 16
-----
Student 2 details:
RollNo : 11
Name : Grisma
Address : chitwan
Age : 15
-----
Student 3 details:
RollNo : 12
Name : Ankit
Address : Butwal
Age : 17
-----
Student 4 details:
RollNo : 13
```



## Exercise

1. What is structure? Write the advantages of using structure.
2. What is union? What are the advantages of using union?
3. Differentiate between structure and union with example.
4. Explain how the value of structure can be accessed.
5. Differentiate between array and structure.

## ■ Programming Exercise

1. Write a program to enter name and marks of 20 students and display them using structure.
2. Write a program to enter the marks of 5 students in Nepali, Computer and using a structure named Marks having elements roll no., name, Nepali\_marks, maths\_marks and Computer\_marks and then display the percentage of each student.
3. Write a program to enter employee name and basic salary of 50 employees and display the record in proper format using structure.

## Pointers

### Definition of Pointer

Pointer is a variable that stores the reference to another variable instead of storing the actual value. A pointer variable contains the address in memory of another variable, object, or function. A pointer is normally declared to be of a specific type depending on what it points to, such as a pointer to a char. The object may be any C data type such as integer, character, string, or structure. The type of a pointer depends on the type of the variable it points to.

### Pointer Variable

If a variable is going to hold a pointer it must be declared. The pointers are declared using the asterisk symbol \* with the data type name and name of the pointer to be declared.

### Syntax to Declare Pointer

```
data_type * pointer_variable_name;
```

#### Example:

```
int *p;
Char *ptr;
```

The data type of pointer and the variable must match, an int pointer can hold the address of int variable, and similarly a pointer declared with float data type can hold the address of a float variable.

- Integer type pointer holds data with 2 bytes
- Floating type pointer holds data with 4 bytes
- Char type pointer holds data with 1 byte

In the above example of pointer variable p and ptr has no value that is we haven't stored an address in it in the above declaration. A pointer initialized in this manner is called a "null" pointer. Suppose now that we want to store in ptr the address of our integer variable k. To do this we use the unary & operator and write:

```
ptr = &k;
```

What the ampersand (&) operator does is retrieve the address of k and copies that to the contents of our pointer ptr. Now, ptr is said to "point to" k.

Pointer variable doesnot store a simple value directly. Instead, a pointer stores a reference to another value. For example, there are two variables: num and numPtr. The simple variable num contains the value 25. The variable numPtr is a pointer which contains a reference to the variable num. Its value is not an int. Its value is a reference to an int.

```
int num=25;
int *numPtr;

numPtr=&num;
```

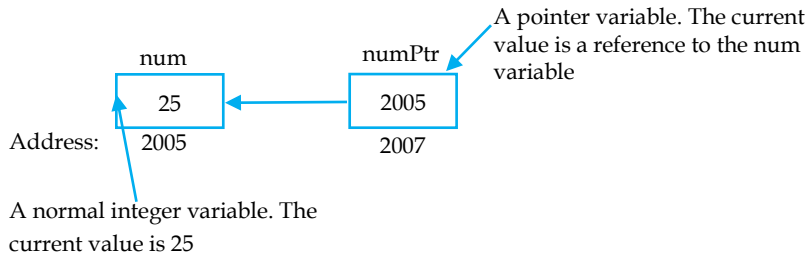


Figure: C pointers

### Benefits of Pointer

- Speed up the execution of program
- Complex data structures like: queues, link lists, stack, trees can be easily implemented.
- Memory can be efficiently utilized
- Hardware interpretation is possible
- Can interact with operating system
- Dynamic memory creation and deletion is possible
- Enables to return more than one value from a function
- Pointers support dynamic memory allocation(DMA)
- Can access memory faster. It is because, accessing the memory location is faster than accessing the value

### Drawbacks of Pointer

- Pointer variable can't be declared as global variable. It is because all the global variables are initially zero (0), which indicates the address 0 of memory.
- Dynamically allocated memory can't be automatically released; it will exist till the end of program. It is programmer's responsibility to release that memory using free () function. This is called memory leak problem.
- If memory is deleted by dynamic memory allocation (DMA) but the pointer that point to the location where it does not exist any object. It is called dangling pointer.

### The NULL Pointer

The constant NULL is a special pointer value which encodes the idea of "points to nothing." It turns out to be convenient to have a well defined pointer value which represents the idea that a pointer does not have a pointee. It is a runtime error to dereference a NULL pointer. In drawings, the value NULL is usually drawn as a diagonal line between the corners of the pointer variable's box...



The C language uses the symbol NULL for this purpose. NULL is equal to the integer constant 0, so NULL can play the role of a Boolean false.

## Address (&) and Indirection (\*) Operator

### Pointer Operator

There are two special pointer operators

#### 1. Address of (&) Operator

The address of operator (&) will return its operand's address. This can be done as follows:

```
int a=5,*p;
p=&a;
```

#### 2. Indirection (\*) Operator

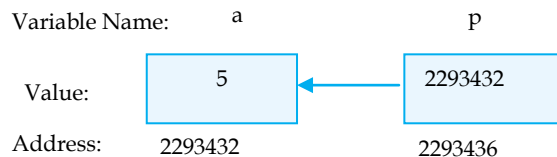
It returns the memory address of the operands. i.e Indirection operator can give the actual value of a variable.

```
int a=5,*p;
p=&a;
```

Value and address can be fetched in following ways:

```
printf("Address of a= %x value of a= %d\n", p, *p);
printf("Address of a=%o value is a= %d\n",&a, a );
```

| Specifier | Meaning                                  |
|-----------|------------------------------------------|
| %x        | Displays the value as hexadecimal number |
| %o        | Displays the value as octal number       |



//Program to illustrate the concept of pointer

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int *p; //pointer variable
    int a=5; //normal variable
    p=&a; //p holds the address of variable a

    printf("Value of a =%d \n", a );
    printf("Value of a =%d \n", *p );
    printf("Address of a=%d\n", p);
    printf("Address of a =%d \n", &a );
    printf("Address of p =%d \n", &p );
    getch();
}
```

**Output:**

```
Value of a =5
Value of a =5
Address of a =2293432
Address of a =2293432
Address of p =2293436
```

## Pointer Expression and Assignment

Several arithmetic operations may be performed on pointers. You can perform arithmetic operations on a pointer just as you can on a numeric value. There are four arithmetic operators that can be used on pointers: ++, --, +, and -.

You can perform the following operations on pointers

- increment(++)
- decrement(--)
- addition of integer to a pointer(+)
- Subtraction of integer from a pointer (-)
- Subtraction of pointer from another pointer
- Comparison of two pointers

You cannot perform the following operations on pointers

- Addition of two pointers
- Multiplication of two pointers
- Division of two pointers

### 1. Incrementing a Pointer

Any pointer variable when incremented (++) points to the next memory location of its type. The pointer gets incremented according to the data type of the value it stores.

**For example:**

```
int *p, a=5;
p=&a;
p++; //increases the address pointed by p by 4 bytes
(*p)++ ; //increases the value of a by 1
```

### 2. Decrementing a Pointer

Any pointer variable when decremented (--) it points to the previous memory location of its type. The pointer gets decremented according to the data type of the value it stores.

**For example:**

```
int *p, a=5;
p=&a;
p--; //decreases the address pointed by p by 2 bytes
(*p)-- ; // decreases the value of a by 1
```

```
// program that shows both Increment and decrement values of a pointer variable
#include <stdio.h>
#include<conio.h>
int main()
{
    int a = 5, *p;
    p = &a;
    printf("Actual value of p= %d\n", p);
    p++;
    printf("Value of p after p++ = %d\n", p);
    p--;
    printf("value of p after p--= %d\n", p);
    getch();
}
```

### Output:

```
Actual value of p= 2293432
Value of p after p++ = 2293436
value of p after p--= 2293432
```

### 3. Addition of Integer to a Pointer

Addition of an integer number to a pointer is allowed. The result is similar to the increment operator with the only difference being the increase in the memory location by the integer number given. The values get incremented according to the type of variable it stores.

```
int *p,a=5;
p=&a;
p=p+2;          //increases the address pointed by p by 8 bytes
```

### 4. Subtraction of Integer from a Pointer

Subtraction of an integer number to a pointer is allowed. The result is similar to the decrement operator with the only difference being the decrease in the memory location by the integer number given. The values get decremented according to the type of variable it stores.

```
int *p,a=5;
p=&a;
p=p-2;          //decreases the address pointed by p by 8 bytes
```

// Program to show both addition and subtraction of integer to a pointer variable

```
#include <stdio.h>
#include<conio.h>
int main()
{
    int x =8 , *p;
    p = &x;
    printf("Actual value of p= %d\n", p);
    p=p+2; // Addition of integer to a pointer
    printf("Value of p after p=p+2 = %d\n", p);
    p=p-2; //Subtraction of integer from a pointer
    printf("value of p after p=p-2= %d\n", p);
    getch();
}
```

**Output:**

```
Actual value of p= 2293432
Value of p after p=p+2 = 2293440
value of p after p=p-2= 2293432
```

**5. Subtraction of One Pointer from Another Pointer**

A pointer variable can be subtracted from another pointer variable only if they point to the same data type. If two pointers are of different data type then type mismatch occurs. This can be used to know how many bytes are free in the memory.

```
int *pa,*pb;
int a=5,b=15,c;
pa=&a;
pb=&b;
c=pa-pb;//gives number of bytes between the address pointed by pa
and pb

//program to illustrate Subtraction of one pointer from another pointer
#include <stdio.h>
#include<conio.h>
int main()
{
    int *pa,*pb;
    int a=5,b=15,c;
    pa=&a;
    pb=&b;
    c=pa-pb;
    printf("pa=%d\n",pa);
    printf("pb=%d\n",pb);
    printf("c=%d\n",c);
    getch();
}
```

**Output:**

```
pa=2293424
pb=2293420
c=1
```

**6. Comparison of Two Pointers**

Comparison of two pointer variables is possible only if the two pointer variables are of the same type. These comparisons are to check equality or inequality. The result is TRUE if both the pointers point to the same location in the memory and FALSE if they point to different locations in the memory.

```
// Program to illustrate comparison of two pointers
#include <stdio.h>
#include<conio.h>
```

```

int main()
{
    int *pa,*pb;
    int a=5,b=15,c;
    pa=&a;
    pb=&b;
    if(pa==pb)
    printf("Both pointer points to same location");
    else
    printf("Two pointers points to different locations");
    getch();
}
    
```

### Output:

```
Two pointers points to different locations
```

## Call by Values and Call by Reference

Functions communicate with each other by passing arguments. We can pass the values in two ways. They are:

- Call by value (Pass by value)
- Call by reference ( Pass by reference)

[Described int the function section of the chapter]

The difference between call by value and call by reference is given below:

| SN | Call by value                                                                | Call by reference                                                                   |
|----|------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 1  | Only the value of the arguments are send to the function                     | Addresses of the arguments are send to the function.                                |
| 2  | Any changes made to the formal arguments do not change the actual arguments. | Any changes made to the formal arguments will be reflected on the actual arguments. |
| 3  | The concept of pointer cannot be used in call by value.                      | The concept of pointer can be used in call by reference.                            |
| 4  | Function cannot return more than one value at a time                         | Function can return more than one value at a time.                                  |
| 5  | Call by value is the default method                                          | Call by reference is not the default method                                         |
| 6  | A copy of variable is passed                                                 | A variable itself is passed                                                         |
| 7  | Original values are not modified                                             | Original values are modified                                                        |

## Solved Problems

### 1. Write a program to enter any two numbers and find their sum using pointers.

```
/*Program to find the sum using pointer */
#include<stdio.h>
#include<conio.h>
int main()
{
    int first, second, *p, *q, sum;
    printf("Enter first Number:");
    scanf("%d",&first);
    printf("Enter Second Number:");
    scanf("%d",&second);
    p = &first; //address of first
    q = &second; //address of second
    sum = *p + *q;
    printf("Sum of entered numbers = %d\n",sum);

    getch();
}
```

Output:

```
Enter first Number:7
Enter Second Number:9
Sum of entered numbers = 16
```

```
/* Program to find sum of two numbers using pointer using call by
reference method */
#include <stdio.h>
#include<conio.h>
float add(float *, float *); //function defination
int main()
{
    float first, second, *p, *q, sum;

    printf("Enter first Number:");
    scanf("%f",&first);
    printf("Enter Second Number:");
    scanf("%f",&second);

    sum = add(&first, &second); //function call

    printf("Sum = %f ",sum);

    getch();
}
```

```
float add(float *x, float *y) //function body
{
    float sum;
    sum = *x + *y;
    return sum;
}
```

## 2. Write a program to display the array elements using pointer

```
/* program to display array elements using pointer */
#include <stdio.h>
#include<conio.h>
int main()
{
    int a[5]={1,2,3,4,5};    //array
    int *p;
    p=a; //pointer variable points to first index of array

    printf("Displaying array elements using pointer\n");
    for(int i=0;i<5;i++)
    {
        printf("\n %d",*p);
        p++;
    }
    getch();
}
```

## 3. Write a program to read n numbers in an array and print the sum of numbers using pointer.

```
/* Program to read n numbers and display thier sum using pointer
*/
#include <stdio.h>
#include<conio.h>
int main()
{
    int a[5],n,i;
    int *p,sum=0;
    p=a;

    printf("Enter how many numbers:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter number %d :",i);
        scanf("%d",&a[i]);
        sum=sum+(*p);
        //printf("\n%d",*p);
    }
}
```

```
        p++;  
    }  
    printf("Sum =%d", sum);  
    getch();  
}
```

Output:

```
Enter how many numbers:7  
Enter number 0 :5  
Enter number 1 :4  
Enter number 2 :1  
Enter number 3 :8  
Enter number 4 :6  
Enter number 5 :3  
Enter number 6 :4  
Sum =31
```

## Exercise

1. What is pointer? Write the advantages of pointer.
2. Define indirection and address of operator.
3. Differentiate between call by value and call by reference.
4. What is NULL pointer?

## ■ Programming Exercise:

1. Write a program to add two values by using pointer.
2. Write a program to show pointer arithmetic.

## Working with Files

### Concept of Data File

A data file is any file containing information, but not code; it is only meant to be read or viewed and not executed. For example, information of students, a letter you write in a word processor and a text file are all considered as data files. We can use C program to read or write files permanently using the concept of data file.

When data is stored using variable, the data is lost when the program exits unless something is done to save it. The input and output operations performed so far were done through screen and keyboard only. After termination of program, all the entered data is lost because primary memory is volatile. If the data that we have entered has to be used later, then it is necessary to keep it in permanent storage device. It is necessary to keep data in the permanent storage because it is difficult to handle the large volume of data by programs. C supports the concept of file through which data can be stored on the disk or secondary storage device. The stored data can be accessed or read whenever required.

File handling in C enables us to create, update, read, and delete the files stored on the local file system through our C program. The following operations can be performed on a file.

- Creation of the new file
- Opening an existing file
- Reading from the file
- Writing to the file
- Deleting the file

### Need for File Handling in C Program

- It helps in permanent storage of the data or information generated after running the program
- Large amount of data can be stored for later use
- Easily transfer the contents of a file from one computer system to another
- It saves a lot of time as you don't have to enter all the data from keyboard.
- You can easily carry the data file from one computer to another.

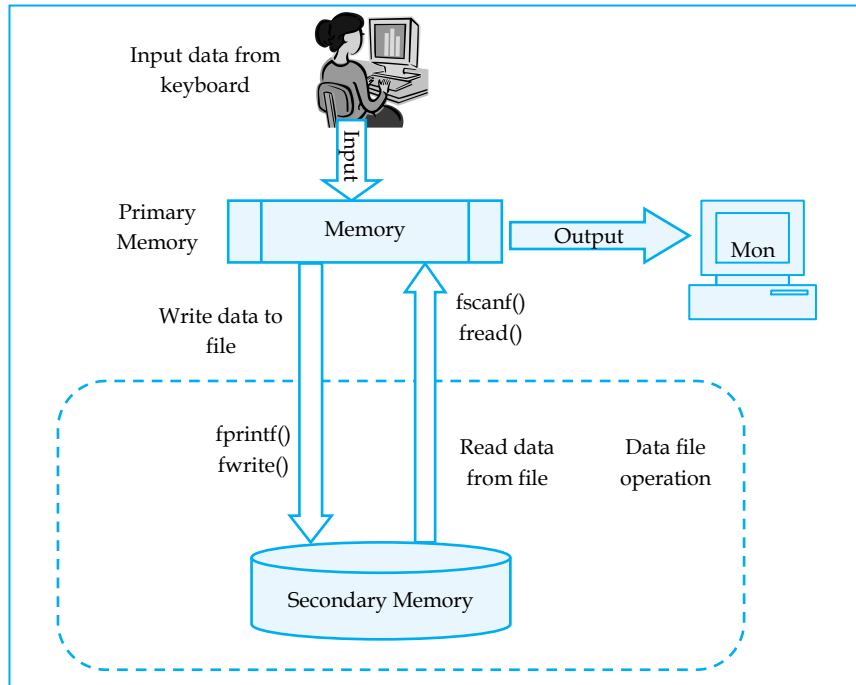


Figure: Concept of data files in c

There are 2 kinds of files in which data can be stored. They are:

- Text files
- Binary files

### Text files:

A Text file contains the text information like alphabets, digits and special symbols which is easily readable. The ASCII codes of these characters are stored in these files. The text file has the `.txt` extension. Text files are often used for files containing plain text that can be opened and viewed with a simple text editor.

### Binary files:

Binary file is the file which can be interpreted and understood by the computer and is not always printable on screen. The binary files has the `.dat` extension. Binary files are often used for all kinds of objects to store data that is not just plain text. In a binary mode data cannot always be interpreted as something printable on screen.

### End of File (EOF):

The file reading function need to know the end of file so that they can stop reading the content from a file. EOF is a sign or a symbol that the end of a file is reached, and that there will be no data anymore. EOF is a condition in a computer operating system where no more data can be

read from a data sources. It is mostly used in character reading functions like `getc()` or `getch()` or `getchar()`. These functions will return a value EOF to indicate that an end-of-file condition has occurred. The value of EOF is mostly -1.

## Sequential and Random File

There are two types of files on the basis of access – the Sequential and the Random access file. The sequential access files allow reading the data from the file in one after another i.e. in sequence. There is no predefined order for accessing data file. All the processes are declare and assigned by the compiler during run time of the program. All the example mentioned in above are sequential access files. The random access files allow reading data from any location in the file. Sometimes, we need to read data file from reverse, middle and from specific location. To achieve this, C defines a set of functions to manipulate the position of the file. The inbuilt functions `fseek()`, `lseek()`, `rewind()` and `ftell()` are the some of the common examples of random access files.

| S.N | Sequential Access                                                                                                                                                                         | Random Access                                                                                                  |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| 1   | Sequential Access to a data file means that the computer system reads or writes information to the file sequentially , starting from the beginning of the file and going on step by step. | Random Access to a file means that the computer system can read or write information anywhere in the data file |
| 2   | The file is accessed slowly                                                                                                                                                               | The file is accessed quickly                                                                                   |
| 3   | Sequential files must search through each and every file                                                                                                                                  | Random files need not to search every files.                                                                   |
| 4   | Insertion and Updating of data is difficult                                                                                                                                               | Insertion and updating of data is easy                                                                         |
| 5   | <code>fread()</code> , <code>fscanf()</code> function is used                                                                                                                             | <code>fseek()</code> , <code>ftell()</code> , <code>rewind()</code> functions are used                         |

## File Manipulation Function

There are many functions in the C library to open, read, write, search and close the file. Lists of file handling functions are given below:

| Functions              | Descriptions                                     |
|------------------------|--------------------------------------------------|
| <code>fopen()</code>   | This function opens new or existing file         |
| <code>fprintf()</code> | This functions write data into the file          |
| <code>fscanf()</code>  | This function reads the data from the file       |
| <code>fputw()</code>   | This function writes an integer value to file    |
| <code>fgetw()</code>   | This function reads an integer value from a file |
| <code>fputc()</code>   | This function writes a character into a file     |
| <code>fgetc()</code>   | This function reads a character from a file      |
| <code>fclose()</code>  | This function closes the file                    |

**fopen():**

fopen() function is used to open a file to perform operations such as reading, writing etc. In a C program, we declare a file pointer and use fopen() as below. fopen() function creates a new file if the mentioned file name does not exist.

**FILE \*fp;**

```
fp=fopen ("filename", "`file opening mode");
```

**Example:**

```
fp=fopen ("book.txt", "r");
```

**fprintf():**

This function writes any types of data (integer, float, string, char etc.) into a file pointed by fp. This function is same as printf function but in spite of writing the data on monitor, it writes the content to the file. It has one extra parameter which points to a file.

**Syntax:**

```
fprintf (fp, "some data");
or
fprintf (fp, "text %d", variable_name);
```

//Program to write "Computer Science" on a file book.txt

```
#include<stdio.h>
#include<conio.h>
int main()
{
    FILE *fp;
    fp=fopen("book.txt", "w");

    fprintf(fp, "Computer Science");
    printf("File successfully written");
    getch();
}
```

**fscanf():**

This function is same as scanf function but it has one extra parameter that points to a file. Instead of reading the data from a standard keyboard, it reads the data from memory.

**Syntax:**

```
fscanf (fp, "format_specifier", variable_name);
```

**Example:**

```
fscanf (fp, "%d", a);
```

//Program to read the book name from a file book.txt

```
#include<stdio.h>
#include<conio.h>
int main()
```

```

{
    FILE *fp;
    char bookname[20];
    fp=fopen("book.txt", "r");

    while (fscanf(fp, "%s", bookname) != EOF)
    {
        printf("Book Name=%s", bookname);
    }
    getch();
}

```

**putw():**

putw function is used to write an integer into a file.

**Syntax:**

```
putw(i, fp);
```

where,

i - integer value

fp - file pointer

**getw():**

getw function reads an integer value from a file pointed by file pointer.

**Syntax:**

```
getw(fp);
```

```
/* Program to illustrate the concept of putw() and getw() */
```

```

#include<stdio.h>
#include<conio.h>
int main ()
{
    FILE *fp;
    int i=1, j=2, num;

    fp = fopen ("aayan.txt", "w");
    putw(i, fp); //writing integer value
    putw(j, fp); //to a file
    fclose(fp);

    fp = fopen ("aayan.txt", "r");

    while (getw(fp) != EOF)
    {
        num= getw(fp); //read an integer value from file
        printf("Data in aayan.txt file is %d \n", num);
    }
    fclose(fp);
    getch();
}

```

**fgetc():**

The `fgetc()` function returns a single character from the file. It gets a character from the stream. It returns EOF at the end of file.

**Syntax:**

```
int fgetc(FILE *stream) ;
```

**Example:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
FILE *fp;
char c;
fp=fopen("abc.txt","r");

while((c=fgetc(fp))!=EOF)
{
printf("%c",c);
}
fclose(fp);
getch();
}
```

**fputc():**

The `fputc()` function is used to write a single character into file. It outputs a character to a stream.

**Syntax:**

```
int fputc(int c, FILE *stream) ;
```

**Example:**

```
#include <stdio.h>
main()
{
FILE *fp;
fp = fopen("abc.txt", "w");//opening file
fputc('a',fp);//writing single character into file
fclose(fp);//closing file
}
```

**fclose():**

`fclose()` function closes the file that is being pointed by file pointer `fp`. To close a file following syntax is used.

```
fclose(file name or file pointer);
```

## Opening, Reading, Writing and Appending Data File

Before you do anything to a file, you need to open a file. After opening a file you can read content from a file, write contents to a file or append content to a file. After read/write is completed, you need to close the file.

## Creating a File

Whenever you want to work with a file, the first step is to create a file. To create a file in a 'C' program following syntax is used:

```
FILE *fp;
```

here,

- FILE is defined in the stdio.h header file
- fp is a pointer variable

## Opening Files

A file must be opened before any I/O operations can be performed on that file. The process of establishing a connection between the program and file is called opening a file. To open a file fopen() function is used as given below:

### Syntax:

```
Pointer _variable = fopen ("file_name.extension", "file mode");
```

### Example:

```
FILE *fp1, fp2;
fp1=fopen("abc.txt", "w");
fp2=fopen("xyz.txt", "r");
fp3=fopen("data.txt", "a");
```

## File Opening Modes

| File Mode | Meaning                                                                                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| r         | "r" for reading<br>This mode opens a file for reading only. The file to be opened must exist. If file is opened successfully fopen() loads it into memory. If a file cannot be opened, fopen() returns NULL.                                            |
| w         | "w" for write<br>This mode opens a empty file for writing only. If the file already exists the previous file will be erased and new file is created in its place. If the file does not exist it will create a new file for writing.                     |
| a         | "a" for append<br>This mode opens a file for appending (i.e. adding the new information at the last). If file does not exist, new file will be created.                                                                                                 |
| w+        | "w+" for (write + read)<br>This mode opens a file for both writing and reading. If file is already exist, previous data will be erased.                                                                                                                 |
| r+        | "r+" for( read + write)<br>This mode opens a file for both reading and writing. The file to be opened must exist and previous data of a file is not erased. This mode is also called update mode because you can add new data and modify existing data. |
| a+        | "a+ for (append + read)<br>This mode opens a file for both reading and appending. A new file is created if the file does not exist. You cannot modify existing data in this mode.                                                                       |

## Closing a File

The files that was opened using `fopen()` function must be closed when no more operations are to be performed on it. After closing a file the connection between file and a program is lost. Although all the files are closed automatically when the program is closed, it is good habit to close a file when there is no more operations to be performed on a file.

### Syntax to close a file

```
fclose(file name or file pointer);
```

You can also close multiple files by calling a function `fcloseall()`. It closes all the opened files.

// Program to open and close a file

```
#include<stdio.h>
#include<conio.h>
int main()
{
    FILE *fp;
    fp=fopen("lovely.txt", "r");
    if (fp==NULL)
    {
        printf("File does not exist !");
    }
    else
    {
        printf("File is opened and ready for I/O operation");
    }
    fclose(fp);

    getch();
}
```

## Solved Problems

### 1. Program to write "Welcome to class 12" in a file

```
#include<stdio.h>
#include<conio.h>
int main()
{
    FILE *fp;

    fp=fopen("class.txt","w");
    fprintf(fp,"Welcome to Class 12");
    printf("Data is written in a file");
    fclose(fp);
    getch();
}
```

**2. Write a program to enter bookname, ISBN no, writername and price of a book and store it in a file book.dat and display in a proper format.**

```

#include<stdio.h>
#include<conio.h>
struct book
{
    char book[25];
    char writer[25];
    int ISBN;
    float price;
}b[100];
int main()
{
    int n,i;
    FILE *fp;
    fp=fopen("book.txt","w");
    printf("Enter how many books:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("Enter Name of book: ");
        scanf("%s",b[i].book);
        printf("Enter ISBN Number: ");
        scanf("%d",&b[i].ISBN);
        printf("Enter Writer Name: ");
        scanf("%s",b[i].writer);
        printf("Enter Price of a book: ");
        scanf("%f",&b[i].price);
        fprintf(fp,"%d\t%s\t%s\t%f \n
",b[i].ISBN,b[i].book,b[i].writer,b[i].price);
    }
    printf("Information successfully stored\n");
    fclose(fp);
    //reading the content from a file
    fp=fopen("book.txt","r");
    printf("ISBN No \t Name of Book \t Writer \t Price \n");

    while(fscanf(fp,"%d %s %s
%f",&b[i].ISBN,b[i].book,b[i].writer,&b[i].price)!=EOF)
    {
        printf("%d \t\t %s\t\t %s\t\t %f
\n",b[i].ISBN,b[i].book,b[i].writer,b[i].price);
    }
    fclose(fp);
    getch();
}

```

**Output:**

```

Enter how many books:2
Enter Name of book: Computer
Enter ISBN Number: 12345
Enter Writer Name: Devendra
Enter Price of a book: 375
Enter Name of book: Accountancy
Enter ISBN Number: 21456
Enter Writer Name: Shreeram
Enter Price of a book: 5.9
Information successfully stored
ISBN No      Name of Book      Writer      Price
12345        Computer          Devendra    375.000000
21456        Accountancy       Shreeram    5.900000

```

**3. Write a program to enter a name of book and append in a file that already exists**

```

#include<stdio.h>
#include<conio.h>
int main()
{
    FILE *fp;
    char bookName[25];
    fp=fopen("book.txt", "a");
    if (fp==NULL)
    {
        printf("File doesnot exist !");
    }
    else
    {
        printf("Enter a book name to append:");
        scanf("%s",bookName);
        fprintf(fp,"%s\n",bookName);
    }
    printf("Successfully appended");
    fclose(fp);
    getch();
}

```

**Output:**

```

Enter a book name to append:Economics
Successfully appended

```

**4. Write a program to count the number of characters, words and lines from a text file**

```

/* Program to count number of characters, words and lines from a text file
*/
#include <stdio.h>
#include<conio.h>
#include <stdlib.h>
int main()
{
    FILE * file;

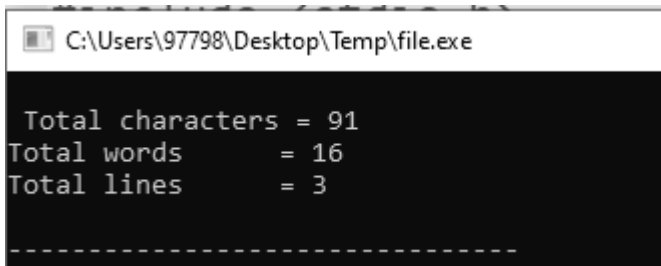
```

```

char ch;
int characters, words, lines;
file = fopen("myfile.txt", "r"); //file myfile.txt must exist

if (file == NULL)
{
    printf("\n Unable to open file ! \n");
    printf("Please check if file exists or not \n");
    exit(EXIT_FAILURE);
}
/* count characters, words and lines */
characters = words = lines = 0;
while ((ch = fgetc(file)) != EOF)
{
    characters++;
    /* Check new line */
    if (ch == '\n' || ch == '\0')
        lines++;
    /* Check words */
    if (ch == ' ' || ch == '\t' || ch == '\n' || ch == '\0')
        words++;
}
/* Increment words and lines for last word */
if (characters > 0)
{
    words++;
    lines++;
}
/* Display records */
printf("Total characters = %d\n", characters);
printf("Total words      = %d\n", words);
printf("Total lines      = %d\n", lines);
fclose(file);
return 0;
}

```

**Output:**


```

C:\Users\97798\Desktop\Temp\file.exe
Total characters = 91
Total words      = 16
Total lines      = 3
-----

```

**5. Write a program to read name and marks of n number of students and store them in a file.**

```
#include <stdio.h>
#include<conio.h>
int main()
{
    char name[50];
    int marks, i, num;
    FILE *fptr;
    fptr = fopen("student.txt", "w");
    if(fptr==NULL)
    {
        printf("Error!");
    }
    printf("Enter number of students: ");
    scanf("%d", &num);
    for(i = 0; i < num; ++i)
    {
        printf("Student %d \n Enter name: ", i+1);
        scanf("%s", name);
        printf("Enter marks: ");
        scanf("%d", &marks);
        fprintf(fptr, "\n Name: %s \n Marks=%d \n", name, marks);
    }
    printf("Students Name and marks successfully written to a
file..");
    printf("\n Press any key to exit..");
    fclose(fptr);
    getch();
}
```

**Output:**

```
Enter number of students: 3
Student 1
Enter name: Kripesh
Enter marks: 75
Student 2
Enter name: Pooja
Enter marks: 76
Student 3
Enter name: Kriti
Enter marks: 84
Students Name and marks successfully written to a file..
Press any key to exit..
```

**6. Write a program to rename a file using rename() function**

```

/* Program to rename a file using rename() function */
#include <stdio.h>
#include<conio.h>
int main()
{
    char oldName[100], newName[100];
    printf("Enter old file Name: ");
    scanf("%s", oldName);
    printf("Enter new file Name: ");
    scanf("%s", newName);
    // rename old file with new name
    if (rename(oldName, newName) == 0)
    {
        printf("File renamed successfully.\n");
    }
    else
    {
        printf("Unable to rename files. Please check files exist
\n");
    }
    getch();
}

```

**Output:**

```

Enter old file Name: student.txt
Enter new file Name: list.txt
File renamed successfully.

```

**7. Write a program to enter any 10 a number from user and store the in a file. Store even numbers in even.txt and odd numbers in odd.txt file.**

```

/* program to store odd and even numbers in a separate files */
#include <stdio.h>
#include<conio.h>
int main()
{
    int num[10],i;
    FILE *fodd;
    FILE *feven;

    fodd=fopen("odd.txt","w");
    feven=fopen("even.txt","w");
    if (fodd==NULL||feven==NULL)
    {

```

```

        printf("Unable to open file !\n");
        printf("Please check whether file exists \n ");

    }
    for(i=0;i<10;i++)
    {
        printf("Enter a number %d:",i+1);
        scanf("%d",&num[i]);
        if(num[i]%2==0)
            fprintf(fodd,"%d \t ",num[i]);
        else
            fprintf(feven,"%d \t ",num[i]);
    }
    printf("The numbers are successfully stored.");

    getch();
}

```

**Output:**

```

Enter a number 1:1
Enter a number 2:4
Enter a number 3:5
Enter a number 4:7
Enter a number 5:6
Enter a number 6:8
Enter a number 7:2
Enter a number 8:3
Enter a number 9:6
Enter a number 10:15
The numbers are successfully stored.
-----
Process exited after 15.33 seconds with return value 0
Press any key to continue . . .

```

**8. Write a program to append any text in an existing file.**

```

/* program to append text in a file */
#include <stdio.h>
#include<conio.h>
int main()
{
    char text[100];
    FILE *fp;
    fp=fopen("myfile.txt","a");
    if (fp==NULL)
    {
        printf("Error!");
    }
}

```

```

    }
    printf("Enter a text to append: ");
    scanf("%[^\n]", text);
    fprintf(fp, " %s ", text);
    printf("Text successfully appended..");
    fclose(fp);
    getch();
}

```

### Output:

```

Enter a text to append: This is appended text
Text successfully appended..
-----
Process exited after 15.9 seconds with return value 0
Press any key to continue . . .

```

## Points to Remember

- A variable is declared by giving it a type and a name (e.g. int k;)
- Once a variable is declared, we can get its address by preceding its name with the unary & operator, as in &k.
- Function can increase the code reusability by avoiding rewriting of same code at multiple locations.
- Functions in C programming can be of two types: Predefined and User defined.
- Predefined functions are defined in a header file.
- User defined function must be explicitly defined by programmer.
- The value or variable that we pass to a function is called parameters. The parameters are of two types: actual and formal parameters.
- Storage class specifies the lifetime and visibility of the variable.
- The advantage of using recursive function is that it makes the code compact and elegant.
- Structure is one which is used to store related fields of different data types.
- Structures are very useful for organizing data in large and complex programs.
- The size of structure is the sum of all data types defined within the structure.
- Dot(.) operator is used to access the element of structure.
- Union is just like structure but it shares a common memory.
- A pointer variable is declared by giving it a type and a name (e.g. int \*ptr) where the asterisk tells the compiler that the variable named ptr is a pointer variable and the type tells the compiler what type the pointer is to point to (integer in this case).

- You can "dereference" a pointer, i.e. refer to the value of that which it points to, by using the unary '\*' operator as in \*ptr.
- An "lvalue" of a variable is the value of its address, i.e. where it is stored in memory. The "rvalue" of a variable is the value stored in that variable (at that address)
- Dynamic Memory Allocation(DMA) is the memory allocated by developer at runtime
- If a program allows you to call a function inside the same function, then it is called a recursive call of the function.
- Structure is usually used when we wish to store dissimilar data together.
- Structure elements can be accessed through a structure variable using dot (.) operator.
- File handling concept is required to store the information in secondary storage devices.
- EOF is not End of File but it is a value returned when to indicate that file has been successfully read and has reached to its last position. If there is any error during file read operation it will return value -1.

### ☑ Multiple Choice Questions

1. The format identifier '%' is also used for \_\_\_\_ data type.
  - a. char
  - b. int
  - c. float
  - d. double
2. Is it possible to run program without main() function?
  - a. yes
  - b. No
  - c. Yes but wrong output
  - d. sometimes
3. How many main() function we can have in our project?
  - a. 1
  - b. 2
  - c. No limit
  - d. Depends on compiler
4. What is the size of an int data type?
  - a. 4 Bytes
  - b. 8 Bytes
  - c. Depends on the system/compiler
  - d. Cannot be determined
5. What is the output of the below code snippet?
 

```
#include<stdio.h>
main()
{
  int x = 5, *p;
  p=&x;
  printf("%u",*p);
}
```

  - a. Address of x
  - b. Address of y
  - c. Address of p
  - d. 5
6. The prototype of a function can be used to,
  - a. Define a function
  - b. Declare a function
  - c. Erase a function
  - d. None of the above
7. Which of the following is a logical OR operator?
  - a. &
  - b. &&
  - c. ||
  - d. None of the above
8. Register variable is .....?
  - a. stored in CPU
  - b. stored in cache memory
  - c. stored in main memory
  - d. stored in secondary memory
9. Automatic variables are stored in \_\_\_\_\_.
  - a. stack
  - b. data segment
  - c. register
  - d. heap
10. What is the format identifier for float data type
  - a. %d
  - b. %c
  - c. %f
  - d. %s
11. Which of the following is true for the static variable?
  - a. It can be called from another function
  - b. It exists even after the function ends
  - c. It can be modified in another function by sending it as a parameter
  - d. All of the mentioned

12. Which of the following are collections of different data types?
- String
  - Structures
  - Char
  - None of the above
13. Union differs from structure in the following way
- All members are used at a time
  - Only one member can be used at a time
  - Union cannot have more members
  - Union initialized all members as structure
14. A function which calls itself is called a \_\_\_ function.
- Self Function
  - Auto Function
  - Recursive Function
  - Static Function
15. The data structure used to implement recursive function calls \_\_\_\_\_
- Array
  - Linked list
  - Binary tree
  - Stack
16. The default parameter passing mechanism is
- Call by value
  - call by reference
  - call by value result
  - None
17. How many times "Hello" is printed?
- ```
int main()
{
while(1)
{
printf("Hello");
}}
```
- 1 time
  - Compilation Error
  - Infinite times
  - Runtime Error
18. A pointer is.....
- A variable that stores address of an instruction
  - A variable that stores address of other variable
  - A keyword used to create variables
  - None of these
19. Is the NULL pointer same as an uninitialised pointer?
- Yes
  - No
  - May Be
  - Can't Say
20. Size of a union is determined by size of the.
- First member in the union
  - Last member in the union
  - Sum of the sizes of all members
  - Biggest member in the union
21. If there is any error while opening a file, fopen will return?
- Nothing
  - EOF
  - NULL
  - Depends on compiler
22. It is not possible to combine two or more file opening mode in open () method.
- TRUE
  - FALSE
  - May Be
  - Can't Say
23. What is the function of the mode 'w+'?
- create text file for writing, discard previous contents if any
  - create text file for update, discard previous contents if any
  - create text file for writing, do not discard previous contents if any
  - create text file for update, do not discard previous contents if any
24. If the mode includes b after the initial letter, what does it indicates?
- text file
  - big text file
  - binary file
  - blueprint text
25. What is the function of fputs()?
- read a line from a file
  - read a character from a file
  - write a character to a file
  - write a line to a file
26. scanf() is a predefined function in\_\_\_\_\_header file.
- stdlib.h
  - ctype.h
  - stdio.h
  - math.h
27. What is the meaning of the following C statement?
- ```
scanf("%[^\n]s", ch);
```
- read all character except new line
  - read all characters
  - read only new line character
  - syntax error
28. What is the following is invalid header file in C?
- math.h
  - mathio.h
  - string.h
  - ctype.h
29. Libray function getch() belongs to which header file?
- stdio.h
  - conio.h
  - stdlib.h
  - stdlibio.h

30. Library function `pow()` belongs to which header file?  
 a. `mathio.h`                      b. `math.h`  
 c. `square.h`                      d. `stdio.h`
31. What is storage class for variable A in below code?  

```
int main()
{
int A;
A = 10;
printf("%d", A);
return 0;
}
```

 a. `extern`                      b. `auto`  
 c. `register`                      d. `static`
32. Can we declare function inside structure of C Programming?  
 a. Yes  
 b. NO  
 c. Depends on compiler  
 d. Yes but runtime error
33. What is the C function used to move current pointer to the beginning of file.? `FILE *fp`;  
 a. `rev(fp)`                      b. `rewind(fp)`  
 c. `rew(fp)`                      d. `wind(fp)`
34. What is the keyword used to declare a C file pointer.?  
 a. `file`                      b. `FILE`  
 c. `FILEFP`                      d. `filefp`
35. Where is a file temporarily stored before read or write operation ?  
 a. Notepad                      b. RAM  
 c. Hard disk                      d. compiler
36. What is the need for closing a file in C language.?  
 a. `fclose(fp)` closes a file to release the memory used in opening a file.  
 b. Closing a file clears Buffer contents from RAM or memory.  
 c. Unclosed files occupy memory and PC hangs when on low memory.  
 d. All the above
37. If a FILE pointer is NULL what does it mean for the following code?  

```
FILE *fp;
fp=fopen("abc.txt", "w");
```

 a. Unable to open a file named `abc.txt`  
 b. `abc.txt` is not available on disk  
 c. Hard disk has hard ware problems.  
 d. All the above
38. A data of the file is stored in.....  
 a. Ram                      b. Hard disk  
 c. Rom                      d. None
39. The content of file will be lost if it is opened in .....  
 a. a mode                      b. w mode  
 c. r mode                      d. a+ mode
40. For a direct access file \_\_\_\_\_  
 a. there are restrictions on the order of reading and writing  
 b. there are no restrictions on the order of reading and writing  
 c. access is restricted permission wise  
 d. access is not restricted permission wise

#### Answer Key

|      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| 1.b  | 2.b  | 3.a  | 4.c  | 5.d  | 6.b  | 7.c  | 8.a  | 9.a  | 10.c |
| 11.b | 12.b | 13.b | 14.c | 15.d | 16.a | 17.c | 18.b | 19.b | 20.d |
| 21.c | 22.b | 23.b | 24.c | 25.d | 26.c | 27.a | 28.b | 29.b | 30.b |
| 31.b | 32.b | 33.b | 34.b | 35.b | 36.d | 37.d | 38.b | 39.b | 40.b |

## Exercise

- What is file handling concept in c? Explain.
- What is the importance of file pointer in file handling? Explain with an example.
- Explain any five file handling functions in c.
- Describe `fscanf()` and `fprintf()` functions with example.
- Define the terms:
  - `FILE`
  - `EOF`

### ■ Programming Exercise

1. Write a program to enter name and salary of employee and write it in a file employee.txt
2. Write a program to enter name, address and salary of 20 employee and store it in a file staff.dat
3. Write a program to enter name, post and salary of a employee and write it in a file employee.dat
4. Write a program which asks name, age, rollno of student and write it in a file student.dat
5. Write a program to display name, age and address reading from a file record.dat.
6. Write a program which reads name, department and age from a file named employee.dat and display them.
7. Write a program which read name, RollNo and age of a student and write them in a file xyz.dat.
8. Write a program which writes "Welcome to Nepal" in a file.

### Project Work

1. Write a menu based program in C language for Library management system which has a following features:
  - a. Inserting a record
  - b. Displaying Record
  - c. Deleting Records
  - d. Updating Records
  - e. Exit







# Object-Oriented Programming (OOP)



## LEARNING OBJECTIVES

After going through this unit, you will be able :

- ◆ Define object oriented concept
- ◆ Define the features of OOP
- ◆ Know about the different applications of OOP concept
- ◆ Able to differentiate between OOP and structured programming language
- ◆ Know the advantages of OOP concept over structured programming

## Programming Paradigms: Procedural, Structural and Object Oriented

### Procedural Programming

A procedural programming language consists of a set of method and a set of code for each method. These languages code programs in such a way that the program executes statement by statement, reading and modifying a shared memory. Thus, we can refer to it as a programming language that executes a set of commands in order. These languages are based on the concept defined by the procedure calls. It means that statements are organized into procedures. These procedures are also known as subroutines or functions. It is written in a step by step manner, thus it is very easy to follow the smaller programs. For simple programming, the procedural approach works fine. However as the program become larger, programmer need a new approach for handling larger programming tasks. Examples of procedural programming includes: Pascal, FORTRAN, COBOL etc.

### Structural Programming

A structured programming also known as modular programming is a programming model that uses the reusable codes. A structural programming language emphasizes on separating a program's data from its functionality. These are based on the top down methodology in which a system is further divided into compositional subsystem. It focuses on dividing the problem into sub parts and hence simplifies the problem solving. The large programs are divided into smaller sub programs so that it can be easily solved. Examples of structural programming includes: ALGOL, Pascal etc.

### Introduction to OOP

Object Oriented Programming (OOP) represents an attempt to make programs more closely model the way people think about and deal with the world. In the older styles of programming, a programmer who is faced with some problem must identify a computing task that needs to be performed in order to solve the problem. OOP approach was invented to overcome the drawbacks of structural programming approach and procedural oriented programming (POP). OOP is the most recent concept in computer programming. Unlike structural programming approach, it uses the bottom up programming approach. OOP treats data as a critical (very important) element in the program development and does not allow data to move freely around the program. OOP ties the data more closely to the function that operates on it and protects data from accidental modification from outside function. OOP allows decomposition of a program into a number of entities called objects and then builds data and functions around these objects. The data of an object can be accessed only by the function associated with that function. However, the function of one object can access the functions of another object.

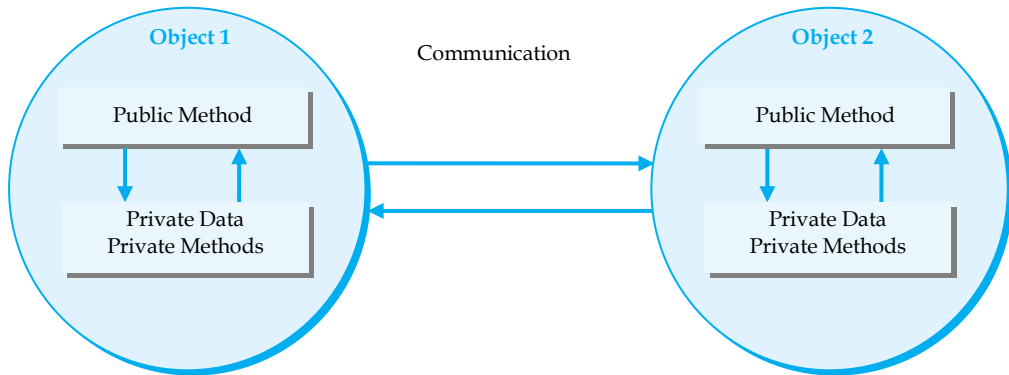


Figure: Objects in Object Oriented Programming

### Benefits of OOPs over Procedural Oriented Programming (POP) are as follows:

OOP offers several benefits to both the programmer and the user. The new technology provides greater programming productivity, better quality of software and lesser maintenance cost. The main advantages of OOP are as follows:

1. **Eliminate redundancy:** Through inheritance, we can eliminate redundant code and extend the use of existing code.
2. **Allows Building Secure Program:** Data hiding principle helps programmer to build secure programs that cannot be accessed by code in other parts of the program.
3. **Ease in division of job:** Since it is possible to map objects of the problem domain to those objects in the program, the work can be easily partitioned based on objects. It is easy to partition the work in a project based on objects
4. **Provide extensibility:** Object oriented systems can be easily upgraded from small to large system.
5. **Reduce complexity:** Software complexity can be easily managed.
6. **Saves development time and increases productivity:** Instead of writing code from scratch, solutions can be built by using standard working modules.
7. **Allows designing simpler interfaces:** Message passing techniques between objects allows making simpler interface descriptions with external systems.

## Features of OOP: Class, Object, Polymorphism and Inheritance

The Objects Oriented programming language supports all the features of procedural programming languages. In addition, it supports some important concepts and terminology which has made it popular among programming methodology.

The important features of Object Oriented programming are:

- Class
- Object

- Inheritance
- Polymorphism
- Data Hiding
- Encapsulation
- Overloading
- Reusability

## Class

Class is a collection of objects of similar type. For example, mango, apple and orange are members of the class 'fruit'. Classes are user-defined data types and behave like the built in data types of the programming languages. Classes are also known as abstract data types (ADT) in OOP because they use the concept of data abstraction.

Thus, a class is a way to bind the data and its associated functions together. It allows the data to be hidden, if necessary, from external use. Generally, a class specification has two parts:

1. Class declaration and
2. Class function definitions

The class declaration describes the type and scope of its members. The class function definitions describe how the class functions are implemented.

The general form of a class declaration is as follows:

| Class Structure                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>class &lt;class_name&gt; {   private:     variable declaration;     function declaration;   public:     variable declaration;     function declaration; }</pre> |

| Class Example                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------|
| <pre>class item {   private:     int number;     float cost;   public:     getdata(int a, float b);     putdata(void); }</pre> |

The class declaration is similar to a **Struct** declaration in C language. The class body contains the declarations of variables and functions. These functions and variables are collectively called class members.

## Object

Object is the basic element of OOP approach which is declared by using class. Object is the basic run time entities in an object-oriented program which consists of data and functions. They may represent a person, a place, a bank account, a table of data or any item that the program has to handle. It may also represent user-defined data. Programming problem is analyzed in terms of

objects. The program objects are similar to the real world objects. Objects in program occupy some space in the memory and have an associated memory address like a structure in C. Hence, an object is a variable consisting both routine and data that is treated as discrete entity within programming.

Each object contains data and functions to manipulate the data. The different objects can interact by sending message through function to one another.

## Encapsulation

Encapsulation is one of the most striking features of OOP approach. As it suggests, the wrapping up of data and function into a single unit called class is known as encapsulation. Unlike the POP approach, the data is hidden inside the object. The data is not accessible to the outside world and only those functions which are wrapped in the class can access the data. This feature of OOP approach prevents the data from the accidental or unnecessary modification. The functions of the object provide the interface between the object data and program. This insulation of the data from direct access by the program is called data hiding or information hiding feature of OOP.

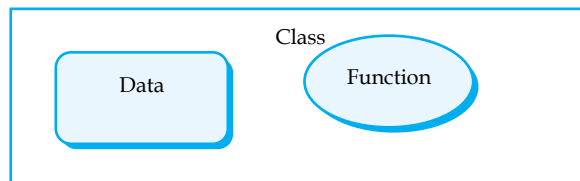


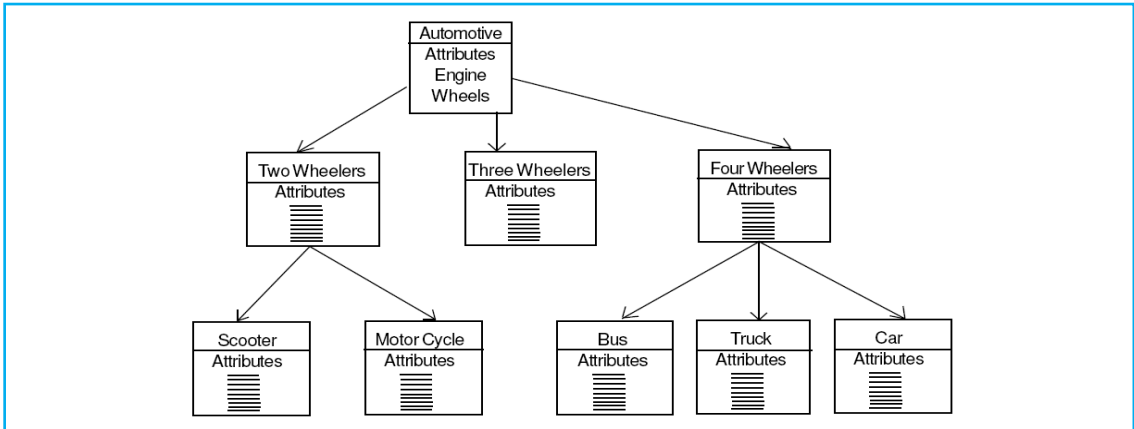
Figure: Encapsulation

## Inheritance

The idea of classes leads to the idea of inheritance when the main class is divided into subclasses, these sub-classes share common characteristics with the class from which they are derived.

Inheritance is the process by which objects of one class acquires the similar properties of objects of another class from which they are derived. For example, 'dog' is a part of class mammal which is again a part of the class 'animal'. Besides it has all the features of the class 'animal', it also has some special features of mammals.

In OOP, the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. This is possible by deriving a new class from the existing one. The new class will have the combined features of both the classes. The following diagram shows an example of inheritance of automobiles.



Inheritance can be classified as follows:

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

### Single Inheritance

In a single inheritance, one class is derived from another class. That is, there is only one base class and only one derived class.

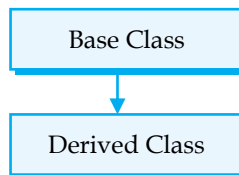


Figure: Single Inheritance

### Multiple Inheritance

In a multilevel Inheritance, one class is derived from multiple classes. In the figure below the class A and class B are base class and class C is derived class.

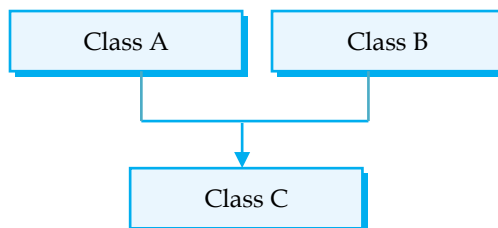


Figure: Multiple Inheritance

## Multilevel Inheritance

In multilevel inheritance, one class is derived from a class which is also derived from another class. It represents a type of inheritance when a derived class is a base class for another class. In a figure below Class B is derived from Class A and Class C is derived from Class B.

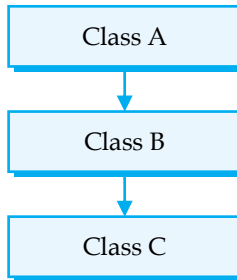


Figure: Multilevel Inheritance

## Hierarchical Inheritance

In Hierarchical Inheritance, there are multiple classes derived from one class. In this case, multiple derived classes allow to access the members of one base class. Here in the figure, Rectangle and Circle are derived class and Shape is base class.

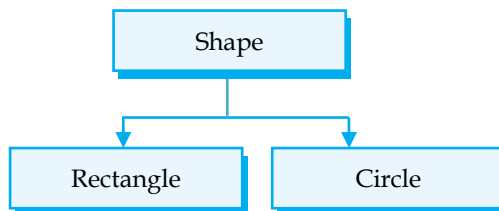


Figure: Hierarchical Inheritance

## Hybrid Inheritance

It is a combination of two or more inheritance. Hybrid inheritance is also known as Virtual Inheritance. In hybrid inheritance derived class have multiple paths to a base class.

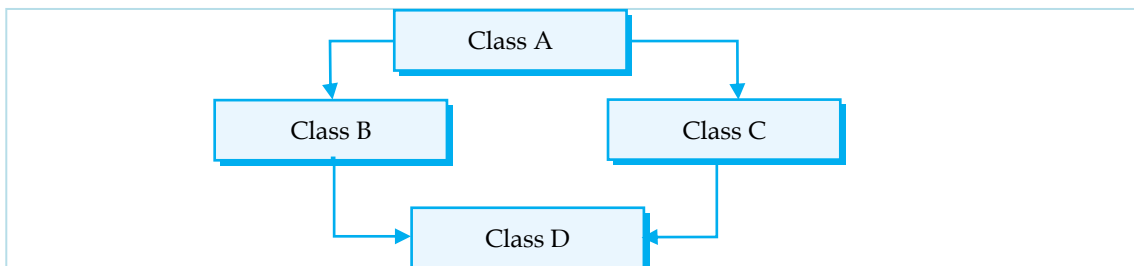


Figure: Hybrid Inheritance

## Polymorphism

Polymorphism means having many forms, In other words polymorphism means the ability of a message to be displayed in more than one forms. An operator or function may exhibit different behaviors (operation) in different instances. The behavior (operation) depends upon the types of data used in the operation. For example, the '+' symbol for two numbers, the operator will generate a sum and if the operands are string (text), then the operator would produce a third string by concatenating two strings. For example 20+30 would return 50 and 'wel'+ 'come' would return 'welcome'. This operator is also used for adding two float numbers as well as float to integer.

The process of making an operator to exhibit different behaviors in different instances is known as operator overloading. Similarly same function can be used for doing different tasks depending upon the context. For example, the function 'draw()' can be used to draw circle, rectangle, square etc. Hence, the function draw() is polymorphic function.

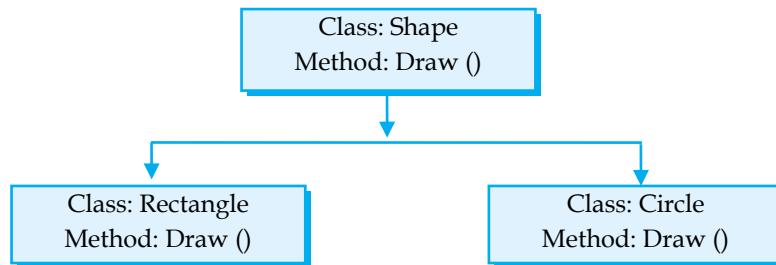


Figure: Polymorphism

The features of OOP can be summarized as follows:

- Emphasis is on data rather than procedures.
- Programs are divided into objects and modules.
- Functions that operate on the data of an object are tied together in the data structure.
- Data is hidden inside the object and cannot be accessed by external functions.
- Objects may communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- Follows bottom-up approach in program design.

## Advantages of OOP

Benefits of OOPs over Procedural Oriented Programming (POP) are as follows:

OOP offers several benefits to both the programmer and the user. The new technology provides greater programming productivity, better quality of software and lesser maintenance cost. The main advantages of OOP are as follows:

1. **Eliminate redundancy:** Through inheritance, we can eliminate redundant code and extend the use of existing code.
2. **Allows Building Secure Program:** Data hiding principle helps programmer to build secure programs that cannot be accessed by code in other parts of the program.
3. **Ease in division of job:** Since it is possible to map objects of the problem domain to those objects in the program, the work can be easily partitioned based on objects. It is easy to partition the work in a project based on objects
4. **Provide extensibility:** Object oriented systems can be easily upgraded from small to large system.
5. **Reduce complexity:** Software complexity can be easily managed.
6. **Saves development time and increases productivity:** Instead of writing code from scratch, solutions can be built by using standard working modules.
7. **Allows designing simpler interfaces:** Message passing techniques between objects allows making simpler interface descriptions with external systems.

## Application of OOP

OOP is a recent method of developing program in computer. Many programming languages such as C++, Java, vb.net etc. have the features of OOP. OOP is suitable for virtually any programming task including development of editors, compilers, databases, communication systems and any other complex real-life application systems.

- Expert system
- Artificial intelligence and virtual reality
- Object oriented database management system
- Decision support system
- Internet and web-based applications
- Mobile computing
- Computer based training and education
- Security system
- Computer aided design and manufacturing
- Image processing and pattern recognitions

## Difference between OOP and Structured Programming Language

OOP is the most recent programming technique whereas Structured Programming is the earlier form of programming and it is also known as Procedural Oriented Programming (POP). The major differences between OOP and Structured Programming Languages are as follows:

| SN | OOP                                                                                         | POP                                                                                     |
|----|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 1  | It is the most recent programming concept which uses the bottom-up approach.                | It is the old programming concept that uses the top-down approach.                      |
| 2  | Programs are divided into a number of entities called objects.                              | Programs are divided into a number of functions.                                        |
| 3  | Emphasis is on data rather than on procedures.                                              | Emphasis is on procedures rather than on data.                                          |
| 4  | It does not allow data to move freely around the program from one object to another object. | It allows data to move freely around the program from one function to another function. |
| 5  | Data is hidden inside the object and cannot be accessed by external functions.              | Data is open and can be freely accessed by all the programs.                            |
| 6  | Problem is viewed as a real world entity.                                                   | Problem is not viewed as a real world entity.                                           |
| 7  | OOP is written by using HLL such as C++, Java, Smalltalk, VB, ASP.NET etc.                  | POP is written by using HLL / MLL such as C, Pascal, FORTRAN etc.                       |
| 8  | The concept of encapsulation provides high level of security                                | Security can't be maintained because it has no such features                            |
| 9  | Easy to reuse existing program codes because of inheritance concept                         | No proper mechanism for reuse of existing program codes.                                |
| 10 | New data and functions can easily be added whenever required                                | Adding new data and function is difficult and time consuming                            |

### Points to Remember

- Object oriented programming (OOP) concept was invented to overcome the drawbacks of structured programming language.
- OOP is able to map the real world problem properly.
- OOP treats data as important element in a program and does not allow data to move freely.
- The data of an object can only be accessed only by the function associated with that object.
- In OOP the programs are divided into objects and modules.
- Object is a real world entity that may represent a place, person, account number or any item that exists.
- Class is a collection of similar objects.
- Encapsulation hides the unnecessary details to the user.
- Concept of Inheritance makes the code reusable.

### ☑ Multiple Choice Questions

1. Which of the following best defines a class?
  - a. Parent of an object
  - b. Instance of an object
  - c. Blueprint of an object
  - d. Scope of an object
2. What is the additional feature in classes that was not in structures?
  - a. Data members
  - b. Member functions
  - c. Static data allowed
  - d. Public access specifier

3. Which is not feature of OOP in general definitions?
  - a. Code reusability
  - b. Modularity
  - c. Duplicate/Redundant data
  - d. Efficient Code
4. Which Feature of OOP illustrated the code reusability?
  - a. Polymorphism
  - b. Abstraction
  - c. Encapsulation
  - d. Inheritance
5. How many classes can be defined in a single program?
  - a. Only 1
  - b. Only 100
  - c. Only 999
  - d. As many as you want
6. Which of the two features match each other?
  - a. Inheritance and Encapsulation
  - b. Encapsulation and Polymorphism
  - c. Encapsulation and Abstraction
  - d. Abstraction and Polymorphism
7. Which among the following best describes polymorphism?
  - a. It is the ability for a message/data to be processed in more than one form
  - b. It is the ability for a message/data to be processed in only 1 form
  - c. It is the ability for many messages/data to be processed in one way
  - d. It is the ability for undefined message/data to be processed in at least one way
8. If same message is passed to objects of several different classes and all of those can respond in a different way, what is this feature called?
  - a. Inheritance
  - b. Overloading
  - c. Polymorphism
  - d. Overriding
9. Which among the following best describes encapsulation?
  - a. It is a way of combining various data members into a single unit
  - b. It is a way of combining various member functions into a single unit
  - c. It is a way of combining various data members and member functions into a single unit which can operate on any data
  - d. It is a way of combining various data members and member functions that operate on those data members into a single unit
10. Which feature can be implemented using encapsulation?
  - a. Inheritance
  - b. Abstraction
  - c. Polymorphism
  - d. Overloading
11. Using encapsulation data security is \_\_\_\_\_
  - a. Not ensured
  - b. Ensured to some extent
  - c. Purely ensured
  - d. Very low
12. Which definition best describes an object?
  - a. Instance of a class
  - b. Instance of itself
  - c. Child of a class
  - d. Overview of a class
13. Which among the following best defines abstraction?
  - a. Hiding the implementation
  - b. Showing the important data
  - c. Hiding the important data
  - d. Hiding the implementation and showing only the features
14. Class is \_\_\_\_\_ abstraction.
  - a. Object
  - b. Logical
  - c. Real
  - d. Hypothetical
15. Which feature of OOP indicates code reusability?
  - a. Encapsulation
  - b. Inheritance
  - c. Abstraction
  - d. Polymorphism
16. Which among the following defines single level inheritance?
  - a. One base class derives another class
  - b. One derived class inherits from one base class
  - c. One base class inherits from one derived class
  - d. One derived class derives from another derived class
17. Which among the following best defines multilevel inheritance?
  - a. A class derived from another derived class
  - b. Classes being derived from other derived classes
  - c. Continuing single level inheritance
  - d. Class which have more than one parent

**Answer Key**

|      |      |      |      |      |      |      |     |     |      |
|------|------|------|------|------|------|------|-----|-----|------|
| 1.c  | 2.b  | 3.c  | 4.d  | 5.d  | 6.c  | 7.a  | 8.c | 9.d | 10.b |
| 11.b | 12.a | 13.d | 14.b | 15.b | 16.b | 17.b |     |     |      |

## Exercise

### ■ Short Answer Questions

1. What is OOP? What are its benefits over structured programming?
2. Differentiate between OOP and structured programming.
3. Explain the terms polymorphism and inheritance.
4. What is object-oriented programming? How is it different from the procedural oriented programming?
5. Why polymorphism and inheritance are important concepts of OOP?
6. What are the advantages of using object oriented programming?
7. What is OOP? Write an application of OOP.
8. Write short notes on:
  - a. Class
  - b. Object
  - c. Polymorphism
  - d. Inheritance

### ■ Long Answer Questions

1. What is object oriented programming? Explain the features of OOP in brief.

## Project Work

1. Visit to one of the IT Company near you and survey which of the programming language they are using to create an application, website or software. List out why they prefer object oriented programming to procedural programming. Also find out the most popular programming technique.





# Software Process Model



## LEARNING OBJECTIVES

After going through this unit, you will be able :

- ◆ Define information system and software project.
- ◆ Know about SDLC and its importance in software development
- ◆ Know about the key difference between system analyst and software Engineer.
- ◆ Know about various methods of requirement collections in SDLC.
- ◆ Know about various types of system design tools such as DFD, ERD, Decision table, Decision tree, Use case etc.
- ◆ Know about various methods of system development
- ◆ Know about documentation and its importance.

## Concept of Software Project

The word 'System' is derived from the Greek word 'Systema' which means '*an organized relationship among components*'. A collection of components or elements that work together to perform a specific task is called system.

You must have heard different types of system even in your own body such as 'Skeleton System', 'Nervous System', 'Muscular System' etc. in which different components are organized in proper way so that they can function a specific task in our body.

Similarly, 'computerized information system' is an organized set of hardware, software and different people to transform given data resources into meaningful and useful information for end users.

Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

A project is well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery). Every project creates a unique product or service.

A Project can be characterized as:

- Every project may have a unique and distinct goal.
- Project is not day-to-day operations.
- Project comes with a start time and end time.
- Project ends when its goal is achieved.
- Project needs sufficient resources in terms of time, manpower, finance, material and knowledge-bank.

The term software project can be defined as the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

## Concept of Software Development Process

A software development process is the process of creating a computer software product. It is a systematic operation that includes designing, preparing the specifications, programming, testing, bug fixing and documentation. These stages are also referred to as the software development lifecycle.

The SDLC provides an international standard that software companies can use to build and improve their computer programs. It offers a defined structure for development teams to follow in the design, creation and maintenance of high-quality software. The aim of the IT software development process is to build effective products within a defined budget and timeline.

## SDLC Life Cycle

Computerized information systems play very important roles in the modern society as it gives us the right information at right time so that we can make right plan for future. All organizations either small or big, private or public, academic institution or business enterprise all need accurate and up to date information for their smooth running in such competitive environment. They have to collect, analyze and process a large volume of data in order to get the valuable information to make right decision for the organization.

Similar to other system, an information system too has a life cycle. SDLC stands for Software Development Life Cycle which consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. Every phase of the SDLC life Cycle has its own process and output that feed into the next phase.

Phases/Stages of SDLC:

- System Study
- System Analysis
- System Design
- System Coding
- System Testing
- System Implementation
- System maintenance and Reviews

### Importance and the Necessity of SDLC

The purpose of SDLC is to specify a set of well-defined steps or phases, coupled with a set of clear, measurable exit criteria, for solving a complex problem (such as developing an information system). The system development life cycle (SDLC) is a set of steps that serves as the basis for most system analysis and design methodologies.

SDLC acts as a memory aid by imposing discipline, thus reducing the risk that key details will be overlooked. Communication is enhanced because the SDLC imposes a consistent set of documentation standards. The steps in the SDLC enhance management control, providing a framework for scheduling, budgeting, and project management. The tools associated with SDLC make it easier to solve the problem. Finally, SDLC increases the possibility that a number of errors are detected early.

The importance of SDLC can be list out as follows:

- It breaks down the entire life cycle of software development process which makes easier to evaluate and debug programs
- It provides guideline to complete each activity systematically for quality system development
- It gives clear idea to developer and save time and cost
- It provides proper documentation so designer can set and use functions, modularity
- It allows to set primarily flexibility and contain a lot of innovation

## Phases of SDLC

In manual system, it is very difficult to collect large volume of data, process them rapidly and acquire specific information necessary for making the right decision. However, these are easily possible in computerized information system. The current information system must be consistent and up-to-date otherwise it has to be updated or replaced with new system.

The process of software development is not an easy task and does not get completed in a single stage. Rather, it is a complex task which requires several stages involving many people who are expert in different areas. These different stages as in life cycle are called SDLC. SDLC is a systematic process of developing a software system. Generally, the SDLC consists of the following stages in order:

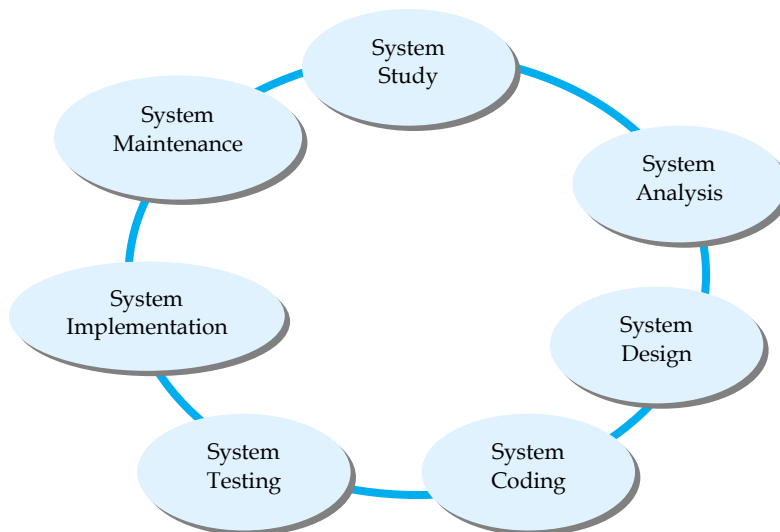


Figure: Stages of SDLC

### 1. System Study

System study is the preliminary investigation or the first phase of system development life cycle. This is the brief study about what actually the physical system is and provides the clear picture of newly developing system. The initial system study involves the preparation of a System proposal which lists the Problem Definition, Objectives of the Study, and Terms of reference for Study, Constraints, Expected benefits of the new system, requirements. The system proposal is prepared by the System Analyst (who studies the system).

In this stage, the development team studies the present system and defines the problems accurately. They interact with the users and gathers information from different sources to recognize the problems of the present system. After collecting the different information of an organization, the system development team proposes the new system with its benefits.

## Feasibility Study

After the development team proposes the new system, the feasibility study or survey is performed in order to determine whether the new system will be feasible or not. The testing is done on the basis of time, cost, technical and operational aspects. After the feasibility study is completed, a report is made which is called 'feasibility survey report' and it is submitted to the management for consideration.

Basically, feasibility study is done on the different aspects based on the system requirements. Some of the widely used feasibility studies are as follows:

- a. **Time Feasibility / Schedule Feasibility:** It is concerned with the time required for the development of new system is feasible for the organization or not. For example, if the organization needs the new system within 6 months and the time required for developing a new system is more than 2 years, then the new system will not be feasible due to time factor.
- b. **Cost Feasibility:** It concerns with the total cost for the development of new system and the returns from the investment in a project. It determines whether the organization can afford the total cost or not. Then it also determines whether it is worthwhile to invest the money for new system or not.
- c. **Technical Feasibility:** It is concerned with specifying different devices and software for the new system. It determines the required devices, which is necessary for the development of new system and also available or not. It searches for the current technology, which includes hardware and software, are able to develop the proposed system or not. If all the technical requirements for the new system can be fulfilled, then the development of new system will be feasible.
- d. **Operational Feasibility:** It is mainly related with human skill and political aspects. The current staffs can work in the new system after training or not. If the whole staffs need very long time and more cost to be trained in the new system, then the new system will not be feasible. But generally the project will not be rejected simply because of operational feasibility.
- e. **Legal Feasibility:** It is mainly focus to analyze any violation of government laws or not. The rules and regulations may bind to develop the current system because of the illegal activities. This feasibility includes foreign money transaction violation, illegal system design, criminal activities on system etc.
- f. **Economic Feasibility:** This feasibility study is performs whether the developing system is economic during the operation time on the client side. Cost-benefit analysis is carried out to analyze the economic feasibility. If operation cost, manpower cost and other costs are within the limit then the new system is considered economically feasible.
- g. **Behavioral Feasibility:** This feasibility study is related to the behavior of the user, person or of the society. The main focus on this feasibility study is how the user response to the new system. It is the behavioral study of the persons in the organization after the arrival of the system. The new system may cause to lose the employment, how employee reacts and other employee related behavior are studied on this feasibility study.

## 2. System Analysis

Systems analysis is a process of collecting accurate data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals.

In this stage, the development team once again goes to the organization and studies very minutely to collect all the drawbacks and details of information from the users, management and data processing personnel. Then the system analyst performs the following activities:

- Gather, analyze, and validate the information.
- Define the requirements and prototypes for new system.
- Evaluate the alternatives and prioritize the requirements.
- Examine the information needs of end-user and enhances the system goal.
- A Software Requirement Specification (SRS) document, which specifies the software, hardware, functional, and network requirements of the system, is prepared at the end of this phase.

## 3. System Design

Based on the user requirements and the detailed analysis of a new system, the new system must be designed. It is the most crucial phase in the development of a system. The logical system design arrived at as a result of system analysis and is converted into physical system design. In the design phase the SDLC process continues to move from the questions of the analysis phase to the how. The logical design produced during the analysis is turned into a physical design - a detailed description of what is needed to solve original problem. Input, output, databases, forms, coding schemes and processing specifications are drawn up in detail. In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided.

There are several tools and techniques used for the system design. These tools are: Flowchart, Data flow diagram (DFD), Data dictionary, Structured English, Decision table and Decision tree.

### Types of System Design

The types of system design are:

- a. **Logical Design:** Theoretically designing of the system is called logical design. The system could be designed on the basis of the requirements. The logical requirement of the system is defined for the further designing of the proposed system. The main idea of this part is just the creation of the logic part of the program.
- b. **Physical Design:** The conversion of logical design into designing tools and techniques is called physical design. It is more detail and complex jobs describing the solution of the

problem. Most of the time it is designed on paper based and sometimes it is designed on computer system by using particular software. It uses algorithms, flowcharts, pseudo codes, decision table, decision tree, E-R Diagram, Data Flow Diagram, context diagram.

#### 4. System Development

After the design stage is complete, the final program specification and the file design are handed over to the programmer. Then the programmers begin to develop the program by using a suitable HLL. In this stage, the programmer has to choose suitable programming language so that the program will be robust and it will be convenient to him. During this stage, the programmer has to detect all the syntax and logical errors and debug them. This stage is considered to be complete when the program can be compiled and executed.

#### 5. System Testing

Once the program modules are ready, each of the program modules is tested independently as per the specifications of the users and debugged. When the modules are bug free, they are integrated as a single system and this system is tested entirely. If the entire system is bug free and can fulfill the requirements, then it is ready to implement.

#### Methods of System Testing

There are two types of testing methods:

1. **White Box Testing/ Glass Box Testing:** The testing is performed with viewing all internal details are called white box testing. In white box testing, the internal source code of a program is tested such as operators, expression, control statement, variables, constant functions etc. Programmer involves doing this job. The testing is done at source level of the software. This type of testing is time consuming and costly.
2. **Black Box Testing/Functional Testing:** The testing is performed without viewing all internal details are called black box testing. Instead of testing the coding part of the software, a program is tested to verify the desired outputs as per the given inputs. Different sample data is used to verify this kind of testing. Any user or person with testing tables can perform this testing. This type of testing is less time consuming and less costly.

#### Types of Software Testing

1. **Unit Testing:** Unit testing performed on each module or block of code during development. Unit Testing is normally done by the programmer who writes the code.
2. **Integration testing:** Integration testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effect on the entire program model.

3. **System testing:** System testing done by a professional testing agent on the completed software product before it is introduced to the market.
4. **Acceptance testing:** Acceptance testing of the product is done by the actual end users.

### System Implementation

After the new system is ready, then it is implemented in the organization. Then the new system becomes a part of the daily activities of an organization. During this stage, application is installed or loaded on existing or new hardware and users are introduced to new system and trained. Installation of the new system can be done in the following three ways.

1. **Direct Implementation:** In this type of implementation, the software is directly installed at user's site by replacing the old system. It is brute one, if the problem persists on the new system the use may face different problems.
2. **Parallel Implementation:** In this type of implementation, both the old and new systems are run in parallel for some time. After monitoring the new system for a reasonable period of time, if it is performing well, then the new system is used by replacing the old one.
3. **Phased Implementation:** In this type of implementation, the system is installed module by module. If one module works efficiently then only another module is installed otherwise modification of first module is performed.

### System Maintenance and Reviews

**Maintenance:** When the system is operating in an organization, users sometimes find problems with how it works and often think of better way to perform its functions. When the time changes, the requirements of the organization also get changes and the system can no longer fulfill it. During maintenance, programmers make changes that users ask for and modify the system to reflect and support changing business condition. These changes are necessary to keep the system running and useful. The maintenance is not a separate phase but a repeated correction of testing phase implements the needed changes.

### Types of Maintenance

There are three types of maintenance, which is used to maintain systems:

1. **Corrective Maintenance:** It corrects the source code of the system for omitting errors. It is mainly used to remove errors as corrective measures. The main aim of this maintenance is to remove bugs for system. Hence, the corrective maintenance is the process of diagnosing and correcting system after they occur.
2. **Adaptive Maintenance:** If the surrounding environment for the system is changed then certain changes should be made. These types of changes in adaptation from rules and regulations changed, policy of organization changed and from other factors is called adaptive maintenance.

- 3. Perfective Maintenance:** It makes the system perfect, up-to-date and improve the life of the system. The maintenance is performed to make the system perfect is called perfective maintenance.
- 4. Preventive Maintenance:** It makes the system prevent form failure in future. The preventive measurement is applied to the system. Its main concern activities are aimed on increasing system maintainability and prevent problems in future.

**Review:** A software review is a process or meeting during which a software product is examined by project personnel, managers, users, customers, user representatives or other interested parties for comment or approval. In this context, software product means any technical document or produced as a software development activity like designs, source code and user documentation.

As a general principle, a technical document is produced showing the progresses and activities involved in software development. The review process is conducted according to market plan, contract signed and on the basis of requirement specifications. Researchers and practitioners have shown the effectiveness of reviewing process in finding bugs and security issues related to software.

## System Analysts | Software Engineer

Both computer system engineers and computer systems analysts do work that is critical for businesses and other organizations. However, they do different types of work and often work in different industries. Computer systems engineers create new types of computer software, and are usually employed by software publishers and electronic device manufacturers. Computer system analysts make sure that computer systems work, and support IT departments in a variety of industries. However there are some different responsibilities of system Analyst and Software Engineer.

Computer system engineers, or systems software developers, work to develop new programs that allow computers to function properly. Unlike applications software developers, systems software developers don't spend their time inventing new games. Instead, they build the operating systems that run computers, such as Windows and Mac OS. Systems software developers also design the interface systems that allow users to interact with programs on the computer. Of course, systems software developers don't only work with computers; they also design systems for mobile phones and video game consoles.

However Computer systems analysts study the specific IT needs of the organization they work for or are consulting for, and take steps to meet those needs either by restructuring or reorganizing an organization's existing resources or by integrating new technology. In certain cases, system analysts may also be involved in training employees how to use new technologies. Some analysts specialize in certain areas of IT analysis. For example, programmer analysts write code to ensure that computer systems work properly, and system Quality Assurance (QA) analysts test an organization's system to search for possible weaknesses.

## Responsibilities of System Analyst

Being the main person in the system development team, the system analyst has to complete many responsibilities. The key responsibility of the system analyst is to coordinate the efforts of all the members of the development team to effectively develop and operate computer based information system. The major responsibilities of a system analyst are as follows:

- 1. Defining Requirements:** Defining requirements means to know the reasons why the users need the new information system. It is the most important and difficult duty of an analyst is to understand the user's requirements. The system analyst may use different fact finding techniques such as interview, questionnaire, field visit, observation etc. in order to define the requirements of the users.
- 2. Prioritizing Requirements:** After the requirements of users have been identified, there is a need to set priority among the requirements of various users. Even the new information system may not fulfill all the requirements of all users at once. The system analyst can prioritize the requirements by having common meeting with all the user sand arriving at a consensus. The system analyst must be able to convince the users about the priority of requirements.
- 3. Analysis and Evaluation:** The system analyst has to find out the drawbacks as well as strength of the current information system. S/he also has to identify and collect the requirements of all the users for new information system. S/he has to analyze the working method of the current information system in the organization and find out its drawbacks and strength. On the basis of the collected information, the system analyst has to evaluate and find the best features of the new information system which will fulfill the user's requirements. The new information system also eliminates the drawbacks of the present system.
- 4. Solving Problems:** System analyst is basically a problem solver as s/he is a team leader in software development process. Being the team leader, s/he has to solve all the problems that may occur during the software development process. The analyst must study the problem in depth and suggest the most appropriate solutions to it.
- 5. Drawing up Functional Specification:** The functional specification refers to the ways of using and handling the new information system in the organization. It is the key duty of system analyst is to obtain functional specifications of the system to be designed. The specification must be non-technical so that even the users and managers understand it.
- 6. Designing System:** Designing the new information system is another main duty of system analyst. After specifications are accepted, the system analyst designs the system. Designing includes two types: logical design and physical design. The design must be understandable to the system implementer. The design must be modular and flexible so that it can easily accommodate changes easily. The system analyst must know the latest design also to assist implementer in his/her task.

7. **Evaluating System:** After the information system has been implemented in the organization, the system analyst must critically evaluate it in order to find out its drawbacks especially for a reasonable period of time. The system analyst must decide when to do evaluation, how to do evaluation, and how to collect user's comments regarding the working methods of the new information system.

## Characteristics of System Analyst

The characteristics/ qualities/ attributes of system analyst are as follows:

1. **Knowledge of organizations:** A system analyst must have the sound knowledge of 'working mechanism', 'management structure' & 'Functional relationship among departments, staffs for which the system should be developed. S/he should understand the organizational and its functional structure, relationship among the departments, daily operations, rules and regulation of the particular system.
2. **Computer system/ Technical Knowledge:** A system analyst must be familiar and well trained in recent relevant areas of computing technologies and enough knowledge about recently developed (updated) system to advice development team which addresses user's need with higher level of efficiency.
3. **Good inter-personal relation:** A system analyst must be good listener, diplomat & able to influence & to resolve conflicting, understand needs & motivates. S/he must be able to resolve conflicting requirements.
4. **Interpersonal Communication skills:** Effective and efficient communication is vital to the success of any team work and system. So, system analyst must be able to talk intelligently with high level management, technical and other staffs and programmers and influence them and change their mind and attitudes.
5. **Analytical skill:** A system analyst must be able to perceive the core problem & discard duplicate data. S/he is required to find solution of problem using appropriate analytical tools
6. **Breadth of knowledge:** A system analyst should have knowledge about various types of peoples, their jobs, perception etc. to handle in team.
7. **Character and Ethics:** Ethics is personal character quality. Successful analyst must be professional, resourceful, inventive and creative. A system analyst requires a strong character and sense of ethics.
8. **Problem Solving Skill:** System is developed to solve the problems. The skills of defining and analyzing the problems, considering, evaluating and selecting the alternatives is known as problem solving skills.

## Requirement collection Methods

Software development life cycle (SDLC) is the process used as the framework for software development. Project managers and business organizations use the SDLC as a blueprint for completing each step of the lifecycle for software development. Each step of the SDLC is called a phase. The requirements gathering and analysis phase is the first phase of the SDLC.

Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for new software being built or modified. Various requirement collection methods are as follows:

- **Interviews:** Interviews of stakeholders and users are critical to creating the great software. Without understanding the goals and expectations of the users and stakeholders, we are very unlikely to satisfy them. We also have to recognize the perspective of each interviewee, so that, we can properly evaluate and address their inputs. Listening is the skill that helps a great analyst to get more value from an interview than an average analyst.
- **Questionnaires or surveys:** Questionnaires, or surveys, allow an analyst to collect information from many people in relatively short amount of time. This is especially helpful when stakeholders are spread out geographically, or there are dozen to hundreds of respondents whose input will be needed to help establish system requirements. When using questionnaires, the questions should be focused and organized by a feature or project objective. Questionnaires should be not be too long, to ensure that users will complete them. When constructing the questionnaire, general guideline to determine the questions would be to ask “how, where, when, who, what, and why.”
- **User Observation:** To get a better understanding of a user in their in current work environment, the analyst may observe the user themselves. User observation is helpful in assisting the analyst by getting a full grasp of how the user interacts with the system, firsthand. When the objective is to improve a task, the analyst can observe the user and how their surroundings affect their interaction with the system. User observation may also be useful in validating data that had been previously collected.
- **Analyzing Existing Documents:** Reviewing the current process and documentation can help the analyst understand the business, or system, and its current situation. Existing documentation will provide the analyst the titles and names of stakeholders who are involved with the system. This will help the analyst formulate questions for interviews or questionnaires to ask of stakeholders, in order to gain additional requirements.
- **Prototyping:** Prototyping is a relatively modern technique for gathering requirements. In this approach, Analyst gather preliminary requirements that he/she use to build an initial version of the solution - a prototype and show this to the client, who then gives additional requirements. Analyst changes the application and cycle around with the client again. This repetitive process continues until the product meets the critical mass of business needs or for an agreed number of iterations.

## Concept of System Design

System design is the construction of a technical, computer based solution for the business requirements identified during system analysis phase. During system design, the system analyst converts the description of the recommended alternative solution into logical and then physical system specification. The different tools used for system design are as follows:

## Data Flow Diagram

The diagram that allows you to model how data flow through an information system, the relationships among the data flows, and how data come to be stored at specific location is known as data flow diagram. It also shows the process that changes or transforms data. Because DFDs concentrate on the movement of data between processes, these diagrams are also called process models.

As the name indicates, DFD is a graphical tool that allows analysts to depict the flow of data in an information system. DFD is a picture of the movement of data between external entities and the processes and data stores within a system.

DFD can be categorized in the following forms:

1. **Context Diagram:** The diagram that shows the scope of the system, indicating which elements are inside and which are outside the system is known as context diagram. A single process represents the whole system in context diagram. It is an overview of an organizational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system. It is also called zero level DFD.

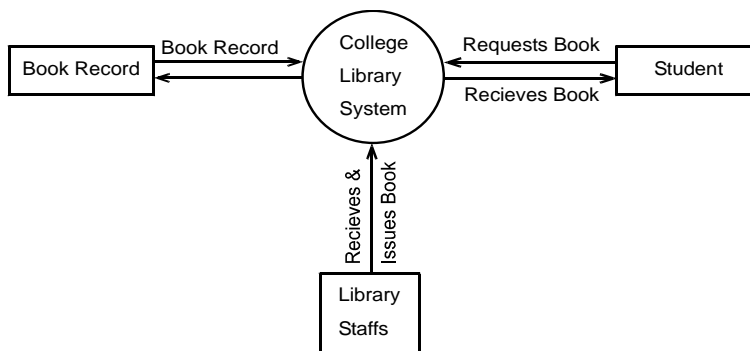


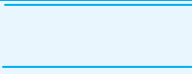

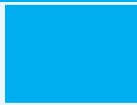

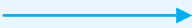
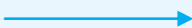


Figure: Context Diagram (Level 0 DFD)

2. **First Level DFD:** A data flow diagram that represents a system's major processes, data flows, and data stores at a high level of details is called First Level DFD. The First Level DFD is just further expansion of Context Diagram. The external entities and external entities are further classified in between data flows of DFD. Hence, it is further refinement of the Context Diagram.

The various symbols used in DFD are as follows:

There are two different standard sets of data flow diagram symbols. Each set consists of four symbols that represent the same things: data flows, data stores, processes, and sources/sinks (external entities).

| Symbols by DeMarco& Yourdon                                                       | Purpose     | Symbols by Gane&Sarwon                                                             |
|-----------------------------------------------------------------------------------|-------------|------------------------------------------------------------------------------------|
|  | Process     |  |
|  | Data store  |  |
|  | Source/Sink |  |
|  | Data Flow   |  |

The key terms used in DFD construction are described as follows:

- Process:** It is the work or actions performed on data so that they are transformed, stored or distributed. During process, the input data is acted upon by various instructions whose result is transformed data. The transformed data may be stored or distributed after finishing processing.
- Data Store:** A data store is data at rest, which may represent one of many different physical locations for data for example, a file folder, one or more computer bases files etc. They are referenced by the process in the system. A data store might contain data about customers, students, customer orders etc.
- Source / Sink:** It refers to the origin and /or destination of data. Sometimes they are referred as external entities as they are outside the system. These external entities may be people, programs, organization or other entities that interact with the system. Once processed, data or information leave the system and go to some other place. The terms source and sink are interchanged with origin and destination respectively.
- Data Flow:** In information system, data should move from one place to another in order to be processed for different purpose. It may be understood as data in motion, moving from one place in a system to another. Data moves in a specific direction from a point of origin to point of destination in the form of document, letter, telephone call or any other medium.

### Rules for Drawing Data Flow Diagram

- For process**
  - No process can have only outputs.
  - No process can have only inputs.
  - A process has a verb phrase label.

## 2. For data store

- Data cannot move directly from one data store to another data store. Data must be moved through a process.
- Data cannot move directly from an outside source to data store. Data must be moved by a process that receives data from the source and places the data into the data store.
- Data cannot move directly to an outside sink from a data store. Data must be moved by a process.
- A data store has a noun phrase label.

## 3. For source/sink

- Data cannot move directly from a source to a sink. It must be moved by a process.
- A source/sink has a noun phrase label.

## 4. For data flow

- A data flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read operation before an update.
- A data flow cannot go directly back to the same process it leaves. There must be at least one other process which handles the data flow, produces some other data flow and returns the original data flow to the beginning process.
- A data flow to a data store means update (delete or change).
- A data flow from a data store means retrieve or use.
- A data flow has a noun phrase label.

## E-R Diagram

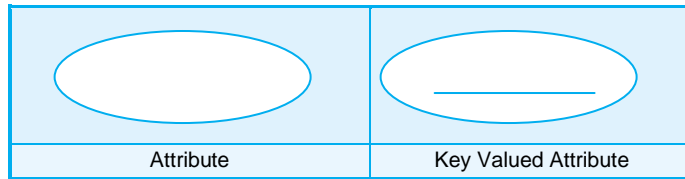
Entity Relationship data model is a technique for detailed and logical representation of the system's data. E-R Diagram is a graphical representation of ER data model. ERD shows the data entities, their attributes and relationship among the entities used in the system.

1. **Entities:** In information system, an entity is a physical or conceptual object like person, place, event, job etc. about which the organization needs to store data. One entity is related to another entity. For example, in a college, student and teacher are two different types of entities but they are linked with each other. Example: student, teacher, school, vehicle etc.

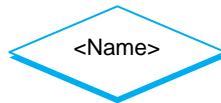
It is denoted by following shape in ER Diagram

**<Entity Name>**

2. **Attribute:** An attribute is a descriptive property or characteristics of an entity. It is also referred by 'element', 'property' or 'field'. Example: attributes of student entity may be Roll no., Name, Address, Photo, Date of birth, Gender etc.



3. **Relationship:** A relationship is a natural business link that exists between one or more entities. The relationship may represent an event that links the entities.



An ERD for 'Customer' and 'Goods' entity

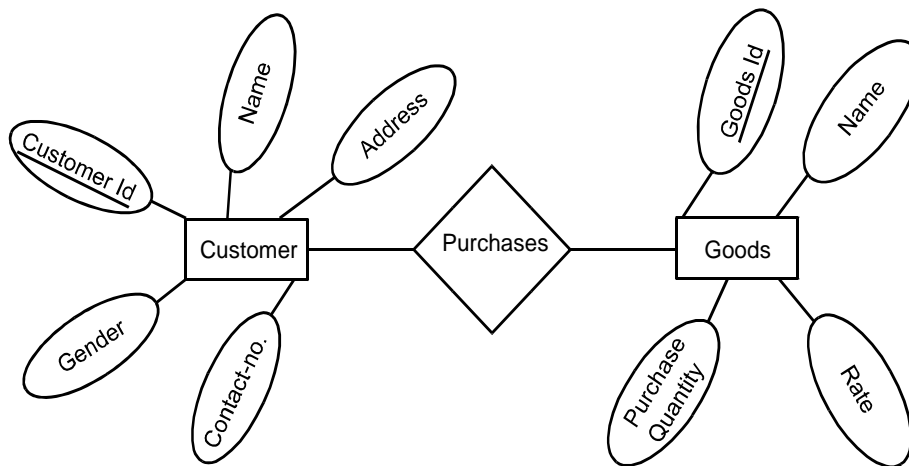


Figure: ERD for Customer and Goods



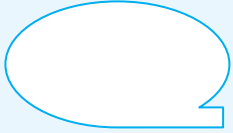



## System Flow Chart

Flowchart is a pictorial representation of an algorithm to solve a particular problem in a computer system. A set of symbols are used for graphical map of control or data path throughout the program. These symbols such as squares, diamonds, circles, rectangles parallelogram and other shapes are used to represent various operations. These symbols are connected by a line with arrowhead to indicate the flow of data or control from one point to another. Flowcharts are used to show the control flow within program execution. Generally, flowcharts are divided into two types: System flowchart and program flowchart.

System flowchart is a type of flowchart that shows the flow of data through a system. This type of flowchart is used to construct working system having memory, storage, input, output and some portion for documentation. This process is the logical design of the working system module. Hence, it explains how a system works by using a diagram.

Some symbols used to draw system flow chart are as follows:

You can find the details about the shapes used in flow chart in Microsoft Word → Insert Tab → Shapes (Under Flowchart Topic) in Office 2007.

| Symbols                                                                            | Purpose             |
|------------------------------------------------------------------------------------|---------------------|
|   | Input/Output        |
|   | Process             |
|   | File stored on tape |
|   | Storage Disk        |
|   | File stored on disk |
|  | Document            |

## Decision Table

Decision table is a technique to represent logic in program development in computer. When several different conditions are involved for specifying complex policies and decision making rules, then decision table becomes very handy.

Decision table is a tabular representation of the logic of a decision, which specifies the possible conditions for the decision and the resulting actions. Decision table uses a standard format and handle combinations of conditions in a very concise manner. There are 3 parts in the decision table: Condition Stub, Action Stub and Rules.

**Condition Stub:** This part of table contains the various conditions that apply in the situation the table is modeling.

**Action Stub:** This part of table lists the actions that result for a given set of conditions.

**Rules:** This part of table specifies which actions are to be followed for a given set of conditions.

| Process Name |       | Rules |  |  |  |  |  |  |
|--------------|-------|-------|--|--|--|--|--|--|
| Conditions   | _____ |       |  |  |  |  |  |  |
|              | _____ |       |  |  |  |  |  |  |
|              | _____ |       |  |  |  |  |  |  |
| Actions      | _____ |       |  |  |  |  |  |  |
|              | _____ |       |  |  |  |  |  |  |
|              | _____ |       |  |  |  |  |  |  |

An example of decision tree to show the calculation of discount policy in a departmental store as follows:

- If the customer is 'Regular' and purchase amount  $\geq 1000$ , then 10% discount.
- If the customer is 'Regular' and purchase amount  $< 1000$ , then 5% discount.
- If the customer is not 'Regular' and purchase amount  $\geq 1000$ , then 5% discount.
- If the customer is not 'Regular' and purchase amount  $< 1000$ , then no (0%) discount.

| Process Name |                                | Rules |   |   |   |
|--------------|--------------------------------|-------|---|---|---|
| Conditions   | Is Customer Regular?           | Y     | Y | N | N |
|              | Is Purchase amount $\geq 1000$ | Y     | N | Y | N |
| Actions      | 10% discount                   | X     | - | - | - |
|              | 5% discount                    | -     | X | X | - |
|              | No discount                    | -     | - | - | X |

## Decision Tree

Decision tree is also a technique to represent condition and actions in a diagrammatic form in computer. Thus, it shows which conditions to consider first and the relationship between each condition and permissible subsequent action to be followed. The diagram resembles the branches on a tree. Hence, it is named decision tree. The root of the tree is the starting point of the decision sequence and progression from the left to right along a particular branch is the result of making a series of decisions.

An example of decision tree to show the calculation of discount policy in a departmental store as follows:

- If the customer is 'Regular' and purchase amount  $\geq 1000$ , then 10% discount.
- If the customer is 'Regular' and purchase amount  $< 1000$ , then 5% discount.
- If the customer is not 'Regular' and purchase amount  $\geq 1000$ , then 5% discount.
- If the customer is not 'Regular' and purchase amount  $< 1000$ , then no (0%) discount.

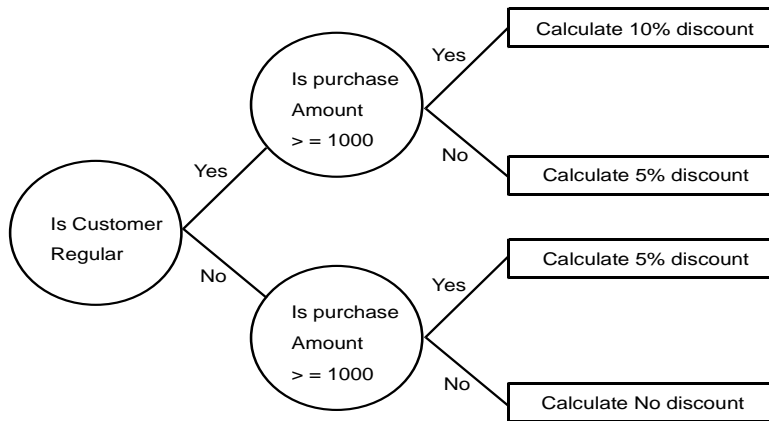


Figure: Decision Tree

## UML

Object Oriented Analysis and Design is a new approach in software development and it is becoming more popular because of its ability to thoroughly represent complex relationships as well as to represent data and data processing with consistent notation. There are wide varieties of techniques such as *Use Cases*, *Class Diagrams*, *State Diagrams* and *Sequence Diagram* that system analyst can use for object oriented analysis and design. All these techniques and associated notations are incorporated into a standard object-oriented language called **Unified Modeling Language (UML)**. Actually, it is the outcome of the efforts of 3 leading computer experts, Grady Booch, Ivar Jacobson and James Rumbaugh who have defined the object oriented modeling language.

Thus, UML is a language for specifying, visualizing, and constructing the artifacts of software system, as well as for business modeling. The UML notation is useful for graphically depicting object-oriented analysis and design models. It not only allows you to specify the requirements of a system and capture the design decisions, but is also promotes communication among key persons involved in the development effort.

## Use Case

Use case is a technique that is incorporated in UML. A use case model is developed in the analysis phase of the object-oriented system development life cycle. It helps the analyst to capture the functional requirements of a system. Thus, it focuses on what the new information system should do rather than how it does. It is done in the early stage of system development to help developers to get a clear understanding of the functional requirements of the new information system without worrying about how these requirements will be implemented. During the requirement analysis stage, the analyst wits with the intended users of the new information system and makes a thorough analysis of what function they desire in the new information system and all these expected functions are represented as use cases.

It consists of **actors** and **actions**. An actor is an external entity that interacts with the system. It is someone or something that exchanges information with the system. An action represents a sequence of related tasks initiated by an actor to accomplish a specific goal. It is specific way of using the system.

The diagram that depicts the use case and actors for a system is called use case diagram.

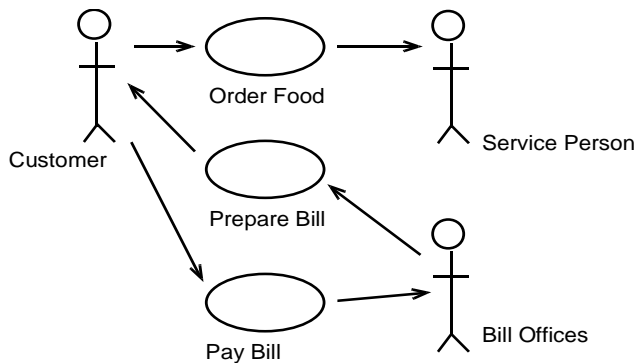


Figure: Use Case between 'Customer' and 'Restaurant'

## SOFTWARE Quality

Quality software refers to software which is reasonably bug or defect free, is delivered in time and within the specified budget, meets the requirements and/or expectations, and is maintainable. The quality of software can be defined as the ability of the software to function as per user requirement. Software quality is a field of study and practice that describes the desirable attributes of software products.

### Software Quality Include

- **Good design:** It's always important to have a good and beautiful design to satisfy users.
- **Reliability:** It should be able to perform the functionality perfectly without issues
- **Durability:** Durability means the ability of the software to work without any issue for a long period of time.
- **Consistency:** Software should be able to perform consistently over platform and devices
- **Maintainability:** Bugs associated with any software should be able to capture and fix quickly and new tasks and enhancement must be added without any trouble
- **Security:** Poor coding and architectural weaknesses often lead to software vulnerabilities. Data must be protected against unauthorized access.
- **Performance:** Performance efficiency refers to an application's use of resources and how that affects its scalability, customer satisfaction, and response times
- **Portability:** Software must be able to work in various environments. The software should be possible to continue using the same basic software in different situations.

## Software Development Model

A software development methodology/model or system development methodology/model in software engineering is a framework that is used to structure, plan and control the process of developing an information system. The different types of most popular system development model are as follows:

### Waterfall Model

The waterfall model is the basis for most systems analysis and design methodologies as it the simplest and the most popular model of SDLC. It is called the waterfall because the model visually suggests work cascading from step to step like a series of waterfalls. It is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall), through the following phases in order:

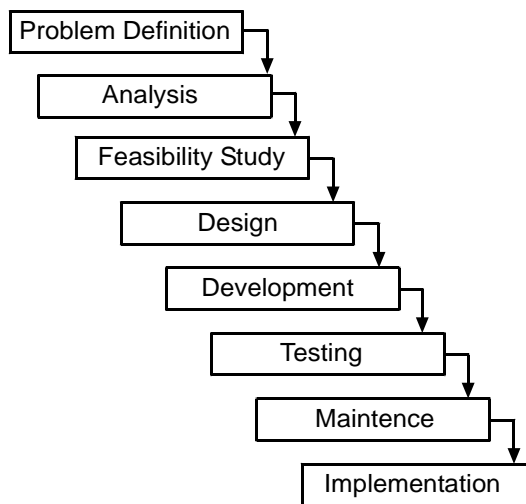


Figure: Waterfall model

In this model, the stages of SDLC are organized in a linear order. The new stage of SDLC begins only after the completion of the previous stage. This model is most suitable for doing routine types of job in program in which all the problems are already known. The documentation is done after the completion of every stage.

Due to its linear order, some people argue that it is a bad idea to follow waterfall model in practice. This is mainly because of their belief that it is impossible for any non-trivial project to get one phase of a software product's lifecycle perfected, before moving on to the next phases.

The fundamental processes of waterfall model are as follows:

1. **Requirements analysis and definition:** It is the first stage of waterfall model. In this stage, the developer should identify the actual requirements of the given problem.

2. **System design:** In this stage the systems design process partitions the requirements to either hardware or software systems.
3. **Implementation and unit system:** During this stage, the system design is realized and tested as a complete system to ensure that the software requirements have been met.
4. **Operation and maintenance:** In this stage, the system is installed to the desire location. The maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system's service as new requirements are discovered.

### Advantages of Waterfall Model

This model is simple and easy to understand and use.

- Simple to use and understand
- It is easy to manage due to the rigidity of the model –each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time .Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

### Disadvantages of Waterfall Model

Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

### Prototype Model

Designing and building a scaled-down but functional version of a desired system is known as prototyping. A prototyping can be built with any computer language or development tools to simplify the process. Prototyping is an iterative process of system development in which requirements are converted to a working system that is continually revised through close work between the system analyst and users.

Software prototyping is the process of creating sample versions of the software program. The creation of a working model of a new computer system or program for testing and refinement is called prototyping. It is used to develop new system and software. This model is much suitable for the development of such new system where there is no clear idea of the requirements, inputs and outputs. Instead of developing a full and complete system, a small prototype (sample) is developed at first. This prototype is discussed with the clients and after getting their feedbacks, the complete system is developed later.

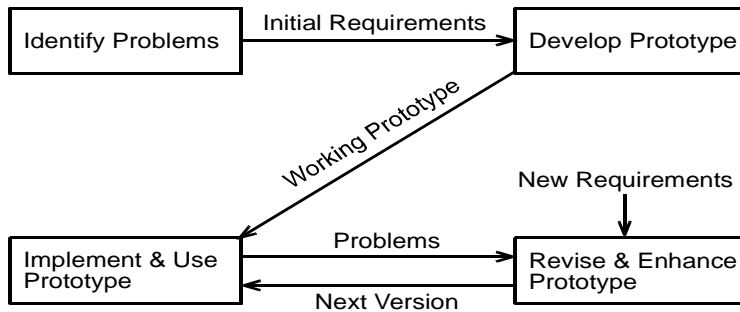


Figure: Prototype Model

Nowadays, prototype can be developed with some 4GL, with query and screen and report design tools of a database management system and with tools called Computer Aided Software Engineering (CASE) tools.

### Advantages of Prototyping

- Communications between the systems analyst and user are improved.
- The expert system can do a better job of determining the user's needs.
- The user plays a more active role in system development.
- The expert and the user spend less time and effort in developing the system.
- Implementation is much easier because the user knows what to expect.
- A system is developed through operational prototyping is capable of easy adaptation.

### Disadvantages of Prototyping

- Because of the iterative nature of prototyping, there is no definite deliverable or competition deadline.
- Code inefficiencies may be a drawback in terms of functionality.
- The computer-human interface provided by certain prototyping tools may not reflect good design techniques.

### Agile Model

"Agile process model" refers to a software development approach based on iterative development. This method break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to

reduce the overall project delivery time requirements. Each and every iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

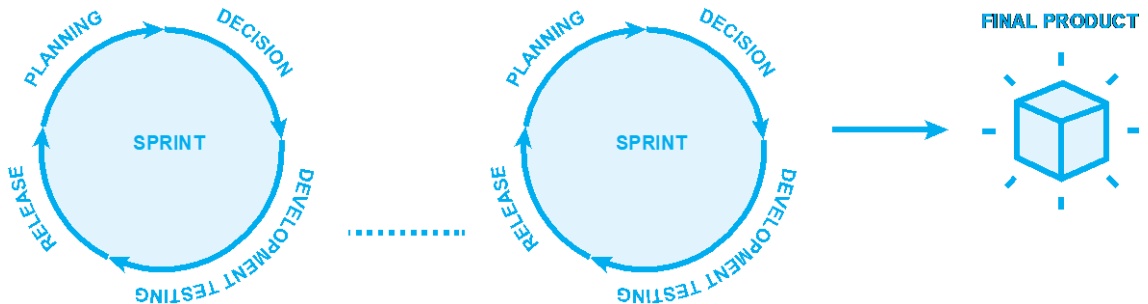


Figure: Agile Development Model

### Phases of Agile Model

Following are the phases in the agile model are as follows:

- Requirements gathering
  - Design the requirements
  - Construction/ iteration
  - Testing/ Quality assurance
  - Deployment
  - Feedback
1. **Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.
  2. **Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.
  3. **Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.
  4. **Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.
  5. **Deployment:** In this phase, the team issues a product for the user's work environment.
  6. **Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

### Advantages of Agile Model

- Frequent Delivery
- Face-to-Face Communication with clients.
- Efficient design and fulfils the business requirement.
- Anytime changes are acceptable.
- It reduces total development time.

### Disadvantages of Agile Model

- Not suitable for handling complex dependencies.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

### Documentation and its Importance

As we know that software development process is not an easy task. Rather it is very complex and time consuming process, it involves many persons who are expert in different areas and it goes through the different stages before the final product is delivered to the user. Documentation plays very important roles in software development process and it provides the basic guide lines for the modification and enhancement of the software in future.

Similarly when the task of one stage is completed, then team members involved in that stage hand over the documentation to the next team members who are involved in another stage so that they can communicate well about the stages of the system.

Documentation may be defined as the process of communicating about the system. It is one of the most important parts of software development. It explains the system. It helps to those who use and maintain software so that it becomes easy to extend, re-design and debug. Documentation is carried out at every state as a part of SDLC. The different tools used in documentation are flow chart, in line comments, data flow diagram, entity relation diagram, class/object diagram etc.

Documentation is a process to help users of software and other people to use and interact with system. Documentation becomes part of each step of system development throughout the process of system development. Software documentation or source code documentation is written text that accompanies computer software. It either explains how it operates or how to use it. Documentation is an important part of software engineering. Document communicates the details about the system targeted at different audience. It explains the system. Hence documentation can be defined as a supporting activity to record information produced by a SDLC process.

Documentation refers to keeping records of all project information in the system development process. The information includes any models of the system components, detailed description of these components as well as information such as the organizational structure and user

responsibilities within this structure. In some systems, it can include the program themselves. The different types of documentation techniques are printed manuals, user manuals or guides, reference manuals, technical reference guides, installation guides, configuration guides, administration guides, online documentation and help system etc.

Similarly documentation plays very key roles for the end user who uses the program for doing the different types of task in their daily life situation as when the end user wants to know more about the system, then the documentation come in use.

#### Documentation is needed because of the following reasons.

- It is a means for transfer of knowledge and details about description of the system.
- To help corporate audits and other requirements of the organization.
- It is needed for IT infrastructure management and maintenance.
- It is needed for migration to a new software platform.

The types of documentation are:

- **Program manual:** It is written by the system programmer during development process. It is written in the line of source code within program. It is very useful for the program modification and maintenance of the system. These types of documentation are not visible for the general users.
- **System manual:** System manual is a physical description of a system, device or process. This technical description is used by expert users and designers as guidelines to maintain and modify various elements of the system. These descriptions are all intended for experts, who must make informed decisions about the installation, capabilities, modifications, and applications of the software.

#### Product Technical Specifications

|                                                                                                                                                            |                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Product name                                                                                                                                               | Mobile phone              |
| Product model                                                                                                                                              | vivo 1908                 |
| Camera                                                                                                                                                     | Front: 5MP<br>Rear: 8MP   |
| Battery                                                                                                                                                    | 4030mAh (TYP)             |
| Dimensions                                                                                                                                                 | 155.11 × 75.09 × 8.28mm   |
| Screen                                                                                                                                                     | 15.80cm (6.22) 1520 × 720 |
| Optimum operating temperature is 0°C to 35°C and optimum storage temperature is -20°C to 45°C. Extreme heat or cold may damage your device or accessories. |                           |

#### SAR Value

|           |                                      |
|-----------|--------------------------------------|
| CE SAR    | 0.869W/kg (Head)<br>1.500W/kg (Body) |
| India SAR | 1.283W/kg (Head)<br>0.609W/kg (Body) |

#### Product Standard Accessories

Cell phone x 1  
Charger x 1  
USB cable x 1  
Phone case x 1

Pin x 1  
Warranty card x 1  
Quick start guide x 1

Version: V1.0 5198876 F372



Figure: System manual of a mobile phone

- **User manual:** User documentation includes the product guidelines addressed to the general user who needs to know basic requirements for getting the best use of software system. User documentation includes the manuals for software product use.

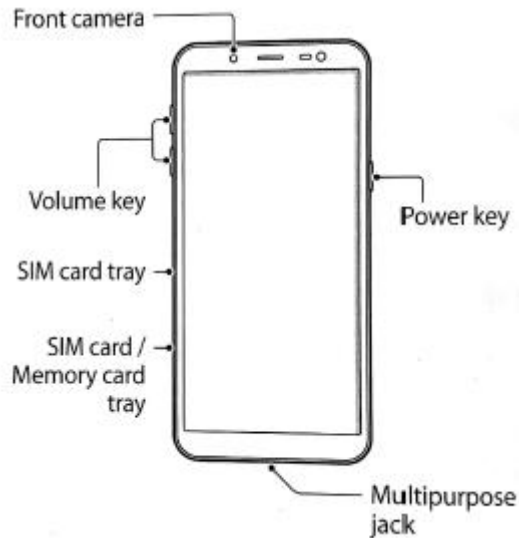


Figure: User Manual of a Mobile Phone

### Points to Remember

- Software is a set of instructions, data or programs used to operate computers and execute specific tasks.
- A software development process is the process of creating a computer software product.
- SDLC provides an international standard that software companies can use to build and improve their computer programs.
- SDLC has seven different stages.
- Feasibility study is performed in order to determine whether the new system will be feasible or not.
- DFD is a graphical tool that allows analysts to depict the flow of data in an information system.
- A software development methodology in software engineering is a framework that is used to structure, plan and control the process of developing an information system.

## ✓ Multiple Choice Questions

1. Which of the following are valid step in SDLC framework?
  - a. Requirement Gathering
  - b. System Analysis
  - c. Software Design
  - d. All of the above
2. The goal of requirement engineering is to develop and maintain sophisticated and descriptive \_\_\_\_\_ document.
  - a. Feasibility Study
  - b. Requirement Gathering
  - c. Software Requirement Validation
  - d. System Requirements Specification
3. It is the process in which developers discuss with the client and end users and know their expectations from the software.
  - a. Requirements gathering
  - b. Organizing Requirements
  - c. Negotiation & discussion
  - d. Documentation
4. It is the process in which developers discuss with the client and end users and know their expectations from the software.
  - a. Requirements gathering
  - b. Organizing Requirements
  - c. Negotiation & discussion
  - d. Documentation
5. Which one is not considered during Technical Feasibility?
  - a. Expandability
  - b. Existence of technology
  - c. Suitability of technology
  - d. Human resource to use the echnology
6. Which of the following is an example of Black Box and Functional Processing?
  - a. First Generation Language
  - b. Second Generation Language
  - c. Third Generation Language
  - d. Fourth Generation Language
7. System study involves
  - a. study of an existing system
  - b. documenting the existing system
  - c. identifying current deficiencies and establishing new goals
  - d. All of the above
8. Feasibility study is carried out by
  - a. managers of the organization
  - b. system analyst in consultation with managers of the organization
  - c. users of the proposed system
  - d. systems designers in consultation with the prospective users of the system
9. Final specifications are drawn up by
  - a. system analyst in consultation with the management of the organization
  - b. the managers of user organization
  - c. system analyst in consultation with programmers
  - d. system designers along with users
10. The final specifications are arrived at:
  - a. after feasibility study
  - b. during feasibility study
  - c. just before implementation phase
  - d. when the system is being designed
11. System design is carried out:
  - a. as soon as system requirements are determined
  - b. whenever a system analyst feels it is urgent
  - c. after final system specifications are approved by the organization
  - d. whenever the user management feels it should be done
12. The primary objective of system design is to
  - a. design the programs, databases and test plan
  - b. design only user interfaces
  - c. implement the system
  - d. find out how the system will perform
13. The primary objective of system implementation is
  - a. to train users to operate the system
  - b. to implement designed system using computers
  - c. write programs, create databases and test with live data
  - d. both a and c
14. During system implementation the following are done
  - a. programs are written and tested with operational data
  - b. user documentation is created and users trained
  - c. the system is tested with operational data
  - d. All of the above

15. System evaluation is carried out
  - a. after the system has been operational for a reasonable time
  - b. during system implementation
  - c. whenever managers of user organization want it
  - d. whenever operational staff want it
16. Systems are modified whenever
  - a. user's requirements change
  - b. new computers are introduced in the market
  - c. new software tools become available in the market
  - d. other similar organization modify these system
17. System analysts have to interact with
  - a. managers of organizations
  - b. users in the organization
  - c. programming team
  - d. all of the above
18. The primary responsibility of a systems analyst is to
  - a. specify an information system which meets the requirements of an organization
  - b. write programs to meet specifications
  - c. maintain the system
  - d. meet managers of the organization regularly
19. The responsibilities of a system analyst include
  - a. defining and prioritizing information requirement of an organization
  - b. gathering data, facts and opinions of users in an organization
  - c. drawing up specifications of the system for an organization
  - d. all of the above
20. The most important attribute of a systems analyst is
  - a. excellent programming skills
  - b. very good hardware designing skills
  - c. very good technical management skills
  - d. very good writing skills
21. Among the attributes of a good systems analyst the following are essential
  - a. knowledge of organization
  - b. analytical mind
  - c. ability to communicate orally
  - d. all of the above
22. Among the attributes of a systems analyst the following are most important
  - a. knowledge of computer systems and currently available hardware
  - b. good interpersonal relations
  - c. broad knowledge about various organizations
  - d. All of the above
23. Changes made periodically to a system, after its implementation, is known as system
  - a. Analysis
  - b. design
  - c. development
  - d. maintenance
24. During what phase, the requirement analysis is performed?
  - a. system design phase
  - b. system development phase
  - c. system analysis phase
  - d. system investigation phase
25. A feasibility study
  - a. includes a statement of the problems
  - b. considers a single solutions
  - c. both (a) and (b)
  - d. a list of alternative solution considered
26. The make-or-buy decision is associated with the \_\_\_\_\_ step in the SDLC.
  - a. Problem/Opportunity Identification
  - b. Design
  - c. Analysis
  - d. Development and Documentation
27. Actual programming of software code is done during the \_\_\_\_\_ step in the SDLC.
  - a. Maintenance and Evaluation
  - b. Design
  - c. Analysis
  - d. Development and Documentation
28. The main ingredient of the report documenting the ..... is the cost benefit analysis.
  - a. System Analysis
  - b. Feasibility Study
  - c. System Analyst
  - d. System Design

**Answer Key**

|      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| 1.d  | 2.d  | 3.a  | 4.b  | 5.d  | 6.d  | 7.d  | 8.b  | 9.a  | 10.a |
| 11.c | 12.a | 13.d | 14.d | 15.a | 16.a | 17.d | 18.a | 19.d | 20.c |
| 21.d | 22.d | 23.d | 24.c | 25.a | 26.b | 27.d | 28.b |      |      |

## Exercise

1. Define the term 'Information System' and 'Software Project'.
2. What is SDLC? Why is it important in system development?
3. What is the difference between system Analyst and Software Engineer?
4. What are the different methods of Requirements Collection in software development? Explain.
5. List out the different phases of SDLC. Explain each of them.
6. What are the different models of system development? Explain them.
7. List out the different phases of SDLC. Explain each of them.
8. What are the different tools used in system design? Describe each of them with suitable example.
9. What is feasibility study? Explain the different levels of feasibility study.
10. Define program logic. Explain different types of program logic tools.
11. What is documentation? Why is it important?
12. What are documentation techniques? Explain with examples.
13. What is feasibility study? Why feasibility study is important in system analysis phase? Explain.
14. What is system analysis? What are the major objectives of system analyst? Explain.
15. Explain about the different testing techniques during the system development.
16. Describe the Agile model to explain the SDLC with advantages and disadvantages.
17. Explain the Prototype model for SDLC with advantages and disadvantages.
18. Explain the concept of system design with different tools.
19. Write short notes on:
  - a. Decision Support System (DSS)
  - b. Waterfall Model
  - c. System Analyst
  - d. Software engineer
  - e. DFD
  - f. ERD
  - g. Use Case
  - h. UML

## Project Work

1. Create a Data Flow Diagram, ER Diagram and Use case diagram for a project- Book Store Management system
2. Visit to any Software company near you and find out among the different software project model which model is used most of the time and why?





# Recent Trends in Technology



## LEARNING OBJECTIVES

After going through this unit, you will be able :

- ◆ Concept of Artificial Intelligence (AI) and Robotics
- ◆ Concept of Cloud Computing
- ◆ Concept of Big Data
- ◆ Concept of Virtual Reality
- ◆ Concept of e-com, e-medicine, e-gov.
- ◆ Concept of Mobile Computing
- ◆ Concept of Internet of things (IoT)

## Concept of Artificial Intelligence (AI) and Robotics

### Artificial Intelligence

Artificial Intelligence is an area of computer science that emphasizes the creation of intelligent machines that work and react like human beings. The Artificial Intelligence field is generally defined as the study and design of intelligent agents where an intelligent agent is a system that recognizes its environment and takes actions that maximize its chances of success. The primary focus of AI is on speech recognition, machine learning, planning, problem solving and many other knowledge-based areas. John McCarthy, one of the pioneers in AI, defines it as "the science and engineering of making intelligent machines". It is a field of computer science and linguistics that studies computer systems that can recognize and react to human language either spoken or written is called Artificial Intelligence.

Completely AI capable system has not come yet. Still the scientists are trying to develop such computers which can perform like human beings. Researchers have used concepts of psychology, physiology, linguistics and computer science to develop AI systems. Even though, some of the AI-based developments are available like Siri from Apple Inc. It is an AI-based intelligent personal assistant. IBM Watson, an AI-based system for answering questions asked in natural language. Sophia (Humanoid Robot) also developed on the basis of AI. One of the major breakthroughs in the field of AI is by Google DeepMind. DeepMind was designed to learn from human and play against top players.

Application areas of AI have been wider and wider day by day. AI is used for game playing, data mining, medical diagnosis and many other areas throughout the technology industry. You can see many great events where AI has been used. For example, on 11 May 1997, Deep Blue became the first computer chess-playing system to beat the world chess champion, Garry Kasparov. In 2005, a Stanford robot won the Grand Challenge by driving autonomously for 131 miles along an unrehearsed desert trail. In March 2016, DeepMind AlphaGo beats Lee Sedol in a five-game match, the first time a computer Go program has beaten a 9-dan professional without handicaps.

### Applications of AI

AI can be used in various fields of computer application to make it more useful and livelier. AI can be used in multiple areas like: Game playing, Speech recognition, Healthcare, e-commerce, cyber security, communications, smart homes, video games, agriculture, automated systems, social media, marketing, education, smart cars etc. Artificial Intelligence and its application are endless. Some of the application areas of AI are:

1. **Game Playing:** AI plays very important roles in playing games so that we can play with computer which think and decide itself like a human player. It makes the game more exciting and livelier. For example, playing chess uses some AI.

2. **Speech Recognition:** AI is very useful in speech recognition so that we can give data and instructions to computer through speech (word) without using keyboard or mouse. We can give more input into computer in less time.
3. **Understanding Natural Language:** The AI helps the computer to understand our natural language like English, Nepali etc. so that we can communicate with computer like a human being.
4. **Computer Vision:** AI can be used to improve the vision of the real world object on computer so that we can view it with 3-Dimension.
5. **Expert System:** It is software which collects the knowledge and decision making roles of the specialists in a particular field and uses it to analyze and decide according to the requirements. MYCIN is an expert system which could diagnose bacterial infection of blood and suggest treatment. The expert system plays very important roles especially in the area of medical science.
6. **Robotics:** It is the branch of engineering devoted to the creation and training of robots. Robotics work within a wide range of fields, such as mechanical and electronic engineering, cybernetics, bionics, and artificial intelligence, all toward the end of endowing their creations with as much sensory awareness, physical dexterity, independence, and flexibility as possible.
7. **Transportation:** Self driving cars are the examples of application of AI in transportation. The AI system collects data from the vehicle's radar, cameras, GPS, and cloud services to produce control signals that operate the vehicle. Moreover, some high-end vehicles come with AI parking systems already. With the evolution of AI, soon a sufficient amount of fully automated vehicles will be seen on most streets. AI in autopilot uses AI technology which is capable of handling takeoff and landing automatically without the pilot involvement.
8. **Marketing:** AI helps create individualized display ads that website visitors want to see. Today every company is in some way an IT company. Hence as a business leader, you need to move with the market and take steps to control this technology to its best possible extent. AI is enabling shoppers to discover associated products whether it is size, color, shape, or even brand.
9. **Cyber Security:** AI can be used to detect vulnerabilities or abnormal user behavior in business applications such as ERP or Financial systems.
10. **Smart Homes:** Various devices like smart locks, smart switches, etc., are increasingly becoming compatible with various devices, and the application of smart homes is becoming more accessible to the general population every day.
11. **Communication:** ChatBots are very popular nowadays for communication with various devices. These devices are used as virtual assistance. Siri, Cortana, and Alexa are some popular ChatBots. These devices use speech recognition system to perform task on human command.

## Robotics

A branch of engineering that is concern to the creation and training of robots. Robotics works within a wide range of fields, such as mechanical and electronic engineering, cybernetics, bionics and artificial intelligence. At the end of their creation, they check all the features like sensory awareness, physical dexterity, independence and flexibility as possible. Hence, robotics is the engineering science and technology of robots, their design, manufacture, application and structural disposition. Robotics is related to electronics, mechanics and software. The electro-mechanical digital devices process and store the information which is directed by the software. The mechanical parts generate a frameset and shape for the robots. And software creates the robots efficient for the execution of jobs for which it is designed.

### Advantages

- Robots can work 24×7
- Robots can increase productivity efficiency and quality of products.
- Robots can work in hazardous environments
- Robots are much more accurate than humans
- Robots can do repetitive task all the time
- Robots do not need any environmental comfort

### Disadvantages

- It is expensive to build and buy robots
- It requires frequent maintenance for running
- People can lose their jobs due to automatic robots in the factories.
- It requires power supply all the times
- Sometimes it may injure humans
- It cannot respond correctly in some emergencies like human beings. It is because robots are not programmed for emergencies.

### Application Areas of Robotics

Application areas of robotics are as follows:

- It is broadly used in industry where humans are at high risk.
- It is used for accuracy cutting and finishing.
- It is used in vehicle and car factory automation.
- Surgeons prefer robotic assistance surgery.
- They are also used for disable people.
- Bomb diffusion, land mine detection, and military operations also use robots.
- It is broadly used in space.
- It can be used as toy for playing.
- It can be used in agriculture industry.

## Concept of Cloud Computing

Cloud Computing refers to manipulating, configuring, and accessing the hardware and software resources remotely. It offers online data storage, infrastructure, and application. It offers online data storage, infrastructure, and application.

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories:

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

There are a lot of companies which provide cloud services. Some of the companies that provide cloud services are:

- Google Cloud
- Amazon Web Services(AWS)
- Microsoft Azure
- IBM Cloud
- Alibaba Cloud

Cloud computing is a popular option for people and businesses for a number of reasons including cost savings, increased productivity, speed and efficiency, performance, and security. Types of Cloud Services includes: Email, Storage, backup, and data retrieval, Creating and testing apps, Analyzing data, Audio and video streaming, Delivering software on demand etc.

## Cloud Components

The different types of cloud components are:

1. **Clients:** Simply the device end user interacts with, to access the cloud to manage their information. These can include desktops, laptops, mobile phones. There are three types of clients:
  - Mobile clients:
    - Clients who use mobile devices such as smartphones.
  - Thin Clients:
    - Some computers that don't have internal HDD which are used to display information you access.
  - Thick Clients:
    - Typical computers which are used to access the internet and connect to the cloud.
2. **Data Centre:** Which is a server or collection of server, housing the application you subscribe to. This can be housed locally where you access it directly or they may reside externally where you access them via the internet.

- Distributed Server:** Like the data centre, these are the servers you access via the cloud but they might be located anywhere around the world. Having servers spread out geographically as a part of storing all in one location. The server provider has more flexibility and increases security. If one server goes down in one location, then the server service still can be access from another location. To you, the cloud subscriber, it's if there is one big server experiencing no issues at all.

## Concept of Big Data

Big data is a combination of structured, semi structured and unstructured data collected by organizations that can be mined for information and used in machine learning projects, predictive modeling and other advanced analytics applications.

Systems that process and store big data have become a common component of data management architectures in organizations. Big data is often characterized by the 3Vs:

- The large volume of data in many environments,
- The wide variety of data types stored in big data systems and
- The velocity, at which the data is generated, collected and processed.

Big data gives tremendous impact on business, ranging from consumers to supply chain operations and companies. Big data applications are management and processing of distributed data, and it can be a new tool for data analysis and visualization. Big data is useful for business organizations to help those gaining deeper insights of customers' habits and behaviors. Because big data is a growing concept, big data is present in a variety of areas. There are many applications of big data despite it being a relatively new thing in the public. For example, by 2011, 235 terabytes were collected by the US Library of Congress and around 30 billion content is shared on Facebook every month. By 2016, it is predicted that around \$7.4m will be spent on data-related initiatives and \$13.8m in enterprises.

## Concept of Virtual Reality

A simulated 3-D environment that a user can experience and manipulate as these were physical. The user sees the environment on display screens, possibly mounted with special pairs of goggles. Special input devices such as gloves, suits and other devices fitted with motion sensors that detect user actions. Virtual reality is a term that applies to computer-simulated environments that can simulate places in the real world as well as in imaginary worlds. Most current virtual reality environments are primarily visual experiences, displayed either on a computer screen or through special stereoscopic displays. But some simulations include additional sensory information such as sound through speakers, movement of the surroundings and lightening effects.

Virtual reality is often used to describe a wide variety of applications associated with immersive, highly visual and 3D environments. The development of CAD software, graphics

hardware acceleration, head mounted displays; database gloves and neatness have helped simulate the visual environment as real one.

### Advantages of Virtual Reality

The advantages of virtual reality are as follows

- Military application for testing, implementing and training.
- Medical field for practitioner purposes.
- It is broadly used in flight training and gaming applications.
- It is mainly used in computer related manufacturing with simulation.
- Virtual reality is also used in motion pictures.
- It also implemented in radio, fine art, music and fictions.

## Concept of e-com, e-medicine, e-government

### e-Commerce

E-commerce (electronic commerce) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the internet. These business transactions occur either as business-to-business (B2B), business-to-consumer (B2C), consumer-to-consumer (C2C) or consumer-to-business (C2B). Ecommerce is often used to refer to the sale of physical products online, but it can also describe any kind of commercial transaction that is facilitated through the internet. E-Commerce has evolved to make products easier to discover and purchase through online retailers and marketplaces. Independent freelancers, small businesses, and large corporations have all benefited from ecommerce, which enables them to sell their goods and services at a scale that was not possible with traditional offline retail.

Ecommerce can be generally classified into following:

- Business to Business(B2B)
- Business to Customer(B2C)
- Customer to Business(C2B)
- Customer to Customer(C2C)

### Business to Business

In a B2B business model, a business sells its product or service to another business through the use of electronic media. B2B transactions generally have a longer sales cycle, but higher order value and more recurring purchases.

### Business to Customer:

The Business to Customer model is the most common business model. B2C transactions generally have shorter sales cycle. B2C business is flourishing because of new technologies like mobile apps, online advertisements through which they can directly meet their customers.

### Customer to Business

Consumer to Business is a reversed pattern of business to consumer. C2B businesses allow individuals to sell goods and services to companies.

### Customer to Customer

Consumer to consumer is a business model that takes care of the commerce transactions among separate individuals. C2C is easy to use because anyone from the general public can sign up and start trading products.

### Advantages

- Availability: Buying and selling 24×7
- Speed of access: Can access from anywhere at any time
- Wide availability: variety of goods and services are available
- Easy accessibility: Easy to start and manage business
- International reach: Go beyond the local market
- Lower cost: Low operational cost
- Personalization and product recommendations
- Quality: Better quality of services
- Reduced paper work

### Disadvantages

- Limited customer service
- Customers can't touch, feel or see the physical product before purchase.
- Sometimes customers have to wait for product delivery
- There can be lack of security
- Hackers can hack the payment system, and account information
- Customers can't try before they purchase products
- Anyone good or bad can easily start the business

### e-Medicine

The use of information and communication technology to provide, enhance or speed up health care services by accessing linking clinics or physicians' offices to central hospitals. This process is helpful to treat patient situated at remote region. This process can include transmitting all the diagnostic documents including X-Ray, prescription, images of examination and other documents to another site. e-Medicine is a rapidly developing application of clinical medicine where medical information is transferred through interactive audiovisual media for the purpose of consulting and sometimes remote medical procedures or examinations. e-Medicine may be as simple as two health professionals discussing in case over the telephone. This process may be as complex as using satellite technology, videoconferencing equipment or real time medical procedures to conduct medical operation by sitting in distinct geographical regions. e-Medicine generally refers to the use of communications and information technologies for the delivery of clinical care

## Advantages of e-Medicine

The advantages of e-Medicine are as follows

- Medication on remote area becomes possible, where access of qualified professionals is not available.
- If equipment available, knowledge sharing at one common point possible by using e-Medicine.
- This facility will be available at home if there is internet facility.
- It saves time
- It helps in reducing medicine errors

## e-Government

e-Government is the use of Information and Communications Technologies to transform the traditional government by making it easily accessible, transparent, effective and accountable to the general people. e-Governance is a network of organizations to include government, nonprofit and private-sector bodies to transform all the services in the form of electronically. In e-governance there are no distinct boundaries to involvement and deed of jobs by the government.

Political, social, economic and technological factors of the nation determine e-Governance. It establishes a relationship between government and citizens. e-Governance provides greater access to government information and services by making the most of the government services online. This will promote citizens to participate by interacting more friendly with government. e-Governance will also increase government accountability by making its operations more transparent which also reduces the opportunities for corruption. e-Governance is a process of transforming government from manual system to electronic digital system. e-Government is not only used in developed countries but also successfully implementing in the developing countries like Nepal. It is a one stop portal such as [www.nepalgov.gov.np](http://www.nepalgov.gov.np), where citizens have access variety of information and services.

## Advantages of e-Governance

- It is easy to access data and information.
- It become one portal for delivering government services.
- It broadly handles Government To Government (G2G), Government To Business (G2B) and Government To Citizen (G2C) data and information services.
- The government services will be made available to the citizens in a convenient, efficient and transparent manner.
- There will be no territory boundaries, not even bounded by geographical regions.

## Concept of Mobile Computing

Mobile Computing is a technology that allows transmission of data, voice and video via a computer or any other wireless enabled device without having to be connected to a fixed physical link. Wireless devices such as laptop, palmtop, PDA's, cell phones, pagers, sensors etc are mobile computing devices. Mobile computing enables us to connect with other at any time from any geographical location.

The main concept involves:

- Mobile communication that refers to the infrastructure put in place to ensure that seamless and reliable communication. These would include devices such as protocols, services, bandwidth, and portals necessary to facilitate and support the stated services.
- Mobile hardware includes mobile devices or device components that receive or access the service of mobility.
- Mobile software is the actual program that runs on the mobile hardware. It deals with the characteristics and requirements of mobile applications.

### Advantages

- Location Flexibility
- Saves Time
- Enhanced Productivity
- Ease of Research
- Entertainment
- Streamlining of Business Processes

### Challenges/Limitations of Mobile Computing

- Low security
- High delay
- Low bandwidth
- Lack of resources
- Network issues
- Change in communication environment

## Concept of Internet of Things (IoT)

The Internet of things (IoT) describes the network of physical objects – “things” – that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.

By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyper

connected world, digital systems can record, monitor, and adjust each interaction between connected things. The physical world meets the digital world – and they cooperate.

### Technologies Made IoT Possible

- Access to low-cost, low-power sensor technology. Affordable and reliable sensors are making IoT technology possible for more manufacturers.
- Connectivity. A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other “things” for efficient data transfer.
- Cloud computing platforms. The increase in the availability of cloud platforms enables both businesses and consumers to access the infrastructure they need to scale up without actually having to manage it all.
- Machine learning and analytics. With advances in machine learning and analytics, along with access to varied and vast amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continues to push the boundaries of IoT and the data produced by IoT also feeds these technologies.
- Conversational artificial intelligence (AI). Advances in neural networks have brought natural-language processing (NLP) to IoT devices (such as digital personal assistants Alexa, Cortana, and Siri) and made them appealing, affordable, and viable for home use.

### Application of IoT

- Create new efficiencies in manufacturing through machine monitoring and product-quality monitoring. Machines can be continuously monitored and analyzed to make sure they are performing within required tolerances. Products can also be monitored in real time to identify and address quality defects.
- Improve the tracking and “ring-fencing” of physical assets. Tracking enables businesses to quickly determine asset location. Ring-fencing allows them to make sure that high-value assets are protected from theft and removal.
- Use wearable’s to monitor human health analytics and environmental conditions. IoT wearable enable people to better understand their own health and allow physicians to remotely monitor patients. This technology also enables companies to track the health and safety of their employees, which is especially useful for workers employed in hazardous conditions.
- Drive efficiencies and new possibilities in existing processes. One example of this is the use of IoT to increase efficiency and safety in connected logistics for fleet management. Companies can use IoT fleet monitoring to direct trucks, in real time, to improve efficiency.
- Enable business process changes. An example of this is the use of IoT devices for connected assets to monitor the health of remote machines and trigger service calls for preventive maintenance. The ability to remotely monitor machines is also enabling new product-as-a-service business models, where customers no longer need to buy a product but instead pay for its usage.

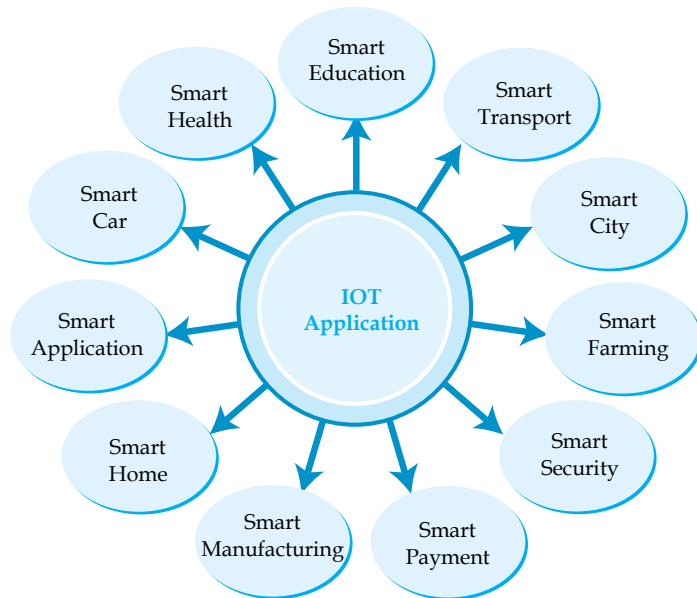


Figure: Applications of IoT

## Challenges to Implement IoT

IoT is complex and poses some risks and is still evolving, many pioneers have started adopting this technology. IoT can help to transform businesses, lives and the world. There are also many opportunities and challenges are going to be as more and more devices start to join the IoT. Some of the common challenges for implementing IoT may be:

- **Connectivity:** Variety of wired and wireless standards are required to enable different application needs.
- **Power:** Many IoT applications need to run for years over battery and reduce the overall energy consumption.
- **Security and Safety:** Protecting user's privacy, manufacturer's IP and blocking malicious activity.
- **Reliability and Stability:** The reliability of sensors used in IoT
- **Data Storage:** Storage of data and information for a long period of time without any damage.
- **Complexity:** IoT application development is very complex and need experts.

## Points to Remember

- Artificial Intelligence field is generally defined as the study and design of intelligent agents
- E-Governance is a network of organizations to include government, nonprofit and private sector bodies to transform all the services in the form of electronically.

- Virtual reality is a term that applies to computer-simulated environments that can simulate places in the real world as well as in imaginary worlds.
- Big data is a combination of structured, semi structured and unstructured data collected by organizations
- Cloud Computing refers to manipulating, configuring, and accessing the hardware and software resources remotely.
- Robotics is a branch of engineering that is concern to the creation and training of robots.

### ✓ Multiple Choice Questions

1. Artificial Intelligence is about \_\_\_\_\_.
  - a. Playing a game on Computer
  - b. Making a machine Intelligent
  - c. Programming on Machine with your Own Intelligence
  - d. Putting your intelligence in Machine
2. The application/applications of Artificial Intelligence is/are:
  - a. Expert Systems    b. Gaming
  - c. Vision Systems    d. All of the above
3. The main function of problem-solving agent is to \_\_\_\_\_.
  - a. Solve the given problem and reach the goal
  - b. Find out which sequence of action will get it to the goal state.
  - c. Both a & b
  - d. None of the above
4. The characteristics of the computer system capable of thinking, reasoning and learning is known is:
  - a. machine intelligence
  - b. human intelligence
  - c. artificial intelligence
  - d. virtual intelligence
5. What is the name of the computer program that simulates the thought processes of human beings?
  - a. Human logic    b. Expert reason
  - c. Expert system    d. Personal information
6. What type of computing technology refers to services and applications that typically run on a distributed network through virtualized resources?
  - a. Distributed Computing
  - b. Cloud Computing
  - c. Soft Computing
  - d. Parallel Computing
7. Cloud computing is a kind of abstraction which is based on the notion of combining physical resources and represents them as \_\_\_\_\_resources to users.
  - a. Real    b. Cloud
  - c. Virtual    d. none of the mentioned
8. Which one of the following cloud concepts is related to sharing and pooling the resources?
  - a. Polymorphism    b. Virtualization
  - c. Abstraction    d. None of the mentioned
9. Which one of the following is Cloud Platform by Amazon?
  - a. Azure    b. AWS
  - c. Cloudera    d. All of the above
10. \_\_\_\_\_is a combination of structured, semi structured and unstructured data collected by organizations that can be mined for information and used in machine learning projects, predictive modeling and other advanced analytics applications.
  - a. Small data    b. Medium data
  - c. Big data    d. None of the above
11. How many V's of Big Data:
  - a. 2    b. 3
  - c. 4    d. 5
12. It is a mediated environment which creates the sensation in a user of being present in a (physical) surrounding.
  - a. WWW    b. VR
  - c. HMD    d. GUI
13. Which of the following describes e-commerce?
  - a. Doing business electronically
  - b. Doing business
  - c. Sale of goods
  - d. All of the above

14. Which of the following is part of the four main types of e-commerce?  
 a. B2B                      b. B2C  
 c. C2B                      d. All of the above
15. Which segment do eBay, Amazon belong.  
 a. B2B                      b. B2C  
 c. C2B                      d. C2C
16. \_\_\_\_ is the use of information and communication technology to provide, enhance or speed up health care services by accessing linking clinics or physicians' offices to central hospitals.  
 a. e-commerce            b. e-governance  
 c. e-medicine             d. All of the above
17. \_\_\_\_ is the use of Information and Communications Technologies to transform the traditional government by making it easily accessible, transparent, effective and accountable to the general people  
 a. e-commerce            b. e-governance  
 c. e-medicine             d. All of the above
18. \_\_\_\_ is a technology that allows transmission of data, voice and video via a computer or any other wireless enabled device without having to be connected to a fixed physical link.  
 a. Cloud Computing    b. Mobile Computing  
 c. e-medicine            d. None of the above
19. Which of the following is the way in which an IoT device is associated with data?  
 a. Internet                 b. Cloud  
 c. Automata               d. Network
20. An IoT network is a collection of \_\_\_\_ devices.  
 a. Signal                  b. Machine to Machine  
 c. Interconnected       d. Network to Network
21. Which of the following is not an application of IoT?  
 a. Arduino                 b. Smart Home  
 c. Smart Car                d. All of the above

#### Answer Key

|      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| 1.b  | 2.d  | 3.c  | 4.c  | 5.c  | 6.b  | 7.c  | 8.b  | 9.b  | 10.c |
| 11.c | 12.b | 13.a | 14.d | 15.b | 16.c | 17.b | 18.b | 19.b | 20.c |
| 21.a |      |      |      |      |      |      |      |      |      |



## Exercise

- What is AI? What are the applications of AI
- What is cloud computing? What are its components?
- What is big data?
- What is virtual reality? What are its advantages?
- What is e governance? Write the advantages of e governance.
- What is mobile computing? Write the advantages of using mobile computing.
- What is IoT? Describe its applications.
- What is IoT? What are the challenges to implement IoT?
- What is e-commerce? Describe its types.

## Project Work

- Cloud Computing and Big data are the recent technologies. Find the company that works on cloud computing and big data. List the companies' names and technology upon which they are working on.
- Make a case study on how e-governance is helping people in the community. Try to visit some government bodies and make a report on how the use of technology is helping to develop a nation.



# References

*Database Systems - Design, Implementation, and Management* by Carlos Coronel, Steven Morris, Reter Rob, Ninth Edition

*Data Communications and Networking, Fourth Edition*, By Behrouz A. Forouzan

*Beginning HTML, XHTML, CSS, and JavaScript*, Jon Duckett,

*The complete reference C*, Herbert Schildt, McGraw-Hill

<https://www.tutorialspoint.com/>

<https://www.w3schools.com/>

*Database Management System*

1. *Concepts of Database Management System*, 2013 Pearson India, Shefali Naik

2. *Database Systems: Design, Implementation, & Management*, 13th Edition, Carlos Coronel

*Concept of Network and Data Communication*

1. *Data Communication and Networking*, Fourth Edition Behrouz A. Forouzan

2. *Data and Computer Communications*, Eighth Edition, William Stallings

*Web Technology II*

1. *Web Technologies: HTML, Javascript, PHP, Java, Jsp, 1. XML and Ajax*, Black Book, Kogent Learning Solutions Inc

2. *Web Technology: A Developer's Perspective*, N. P. Gopalan

*Programming in C*

1. *C Programming Language*, 2nd Edition, Kernighan Brian W., Ritchie Dennis

2. *Programming in ANSI C*, 4th Edition, E Balagurusamy

3. <https://www.javatpoint.com/>

*Object-Oriented Programming (OOP)*

1. *C++ Primer*, 5th Edition, Stanley Lippman, Josée Lajoie, Barbara Moo

2. *Effective Modern C++*, Scott Meyers

*Software Process Model*

1. *New Trends in Software Process Modelling*, 2006 Edition, Silvia T. Acuna, Maria I. Sanchez-segura

2. *Software Process Modeling*, 2005th Edition, Silvia T. Acuna, Natalia Juristo

*Recent Trends in Technology*

<https://azure.microsoft.com>

