COMPUTER SCIENCE

Grade: XII

DATABASE MANAGEMENT SYSTEM (DBMS)

REFERENCE NOTE

NEB Important Questions for Computer Science XII

Unit-1- Database Management System - 8 Marks

1. *What is Database and DBMS? List out the advantages and disadvantages of DBMS.*
2. Explain the different models of DBMS with advantages and disadvantages.
3. ***Write differentiate between centralized and distributed database system.***
4. ***Who is DBA? What are the major responsibilities of DBA?***
5. ***What is normalization? Explain normalization process with examples.***
6. Define the following terms. a) Data Dictionary b) ***Primary Key*** c) Relationship d) DML ***e) SQL*** f) ***Data Integrity*** g) DDL h) ***Data Security*** ***i) Database System***

Unit 1 – Database Management System (DBMS)

Data:

Data is defined as the raw facts and figures. It could be any numbers, pictures, sound, alphabets or any combination of it. Which do not provide clear meaning. Examples, 101, Hari, Bharatpur etc.

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

Data Processing:

Data processing is the mechanism of converting unprocessed data into meaningful result or information.

Information:

When data are processed using a database program or software, they are converted to the meaningful result, called information. Information provides answers to "who", "what", "where", and "when" questions. Examples, Hari lives in Bharatpur-11, Chitwan.

Data Processing

Difference between Data and Information

S.N.	Data	S.N.	Information
1	It is raw or known facts.	**1**	It is processed or refined data.
2	It stores the facts.	**2**	It presents the facts.
3	It is inactive in nature (they exist)	**3**	It is active in nature (It enables doing)
4	It is technology-based	**4**	It is business based
5	Data is gathered from various sources.	**5**	Information is transformed from data
6	Data do not have fixed format.	**6**	Information normally in the form of table, graph, curve line etc.

Flat File/ File based system:

- It is traditional way to keeping records of any organization in a manual filing system. It means to used to keep records in file based or flat file system non computerized.
- A flat file system is a system of flies in which every file in the system must have a different name.

Limitation of file based/ Flat file system

i. Duplication of data (Data Redundancy)
ii. Inconsistent data.
iii. Program Data dependence.
iv. Poor data control.
v. Limited data sharing.
vi. Security problems.
vii. Incompatible file formats.
viii. Fixed queries

Database System:

A database system consists of a collection of interrelated data and a set of application programs to access, update and manage the data.

Database:

It is organized form of record about some person, organization or something store under certain media. It is a collection of related information about a subject organized in a useful manner that provides a base or foundation for procedure, such as retrieving information, drawing conclusion and make decision.

Advantage of database over flat file or file based system

1. Reduction of data redundancies
2. Shared data
3. Data independent
4. Improved integrity
5. Efficient data access
6. Multiple user interface
7. Improved security
8. Improved backup and recovery
9. Supports for concurrent transactions
10. Unforeseen queries can be answered

File based system Vs Electronic Database System

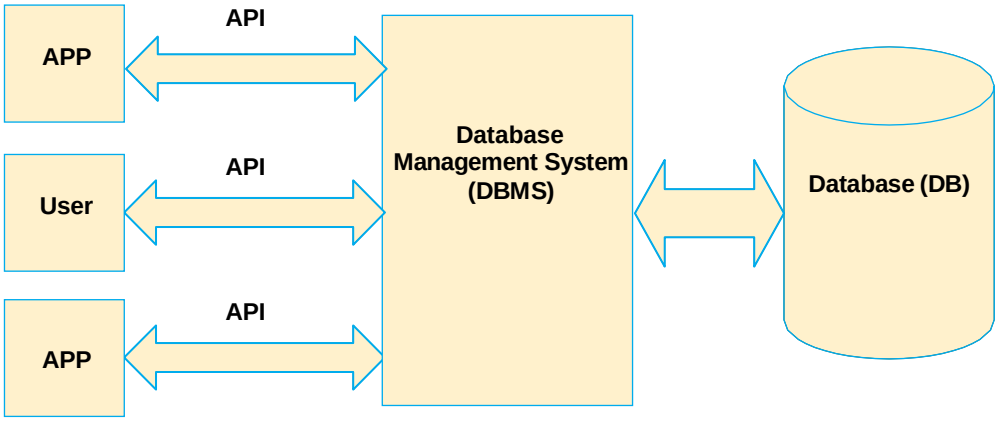
S.N.	File Based System	S.N.	Electronic Database System
1	It provide detail of the data representation and storage of data	1	Database System gives abstract view of data that hides details.
2	It doesn't have a crash recovery mechanism.	2	It provides crash recovery mechanism using backup and other security measures.
3	It is difficult to reduce data redundancy.	3	Data redundancy can be done easily.
4	Searching of data requires a lot of time and effort.	4	Data can be easily searched.
5	Difficult to maintain the database.	5	Easy to maintain the database.

DBMS:

Database Management System is software that manages the data stored in a database. This is a collection of software which is used to store data, records, process them and obtain desired information. Since, data are very important to the end users, we must have a good way of managing data.

A DBMS is a collection of programs that manages the database structure and controls access to the data stored in the database. The DBMS make it possible to share the data in the database among multiple applications or users. The DBMS stands between the database and the user.

Examples: MS-Access, Oracle, FoxPro, dBase, SQL server, MySQL, Delphi, Sybase, etc.



Why to Use DBMS?

1. To develop software application in less time.
2. Data independence and efficient use of data.
3. For uniform data administration.
4. For data integrity and security.
5. For concurrent access to data, and data recovery from crashes.
6. To use user friendly declarative query language.

Some major database System activities are (Functions of DBMS)

1. Adding new file to the database
2. Inserting data into the database
3. Retrieving/viewing data from the database
4. Updating data in existing database file
5. Deleting data from the database file
6. Removing files from the database

Advantages of DBMS (Features /Objectives of DBMS)

1. Sharing data
2. Reduced data redundancy
3. Data backup and recovery
4. Inconsistency avoided
5. Data integrity
6. Data security
7. Data independence
8. Multiple user interfaces
9. Process complex query

Disadvantages of DBMS

1. Expensive
2. Changing Technology
3. Needs Technical Training
4. Backup is needed

Field/ Attribute:

A field is a piece of information about an element. A field is represented by a column. Every field has got a title called the field title.

Record (Tuple):

A record is information about an element such as a person, student, an employee, client, etc. A record can have much information in different heading or titles.

Table:

A table is the arrangements of rows and columns. Each table must have unique name and must be simple. It is the place where data and information are stored.

Table: Student Information		
Roll No.	***Name***	***Address***
1	Gita	Narayangarh
2	Sita	Gaindakot
3	Arjun	Bharatpur

Objects:

Database Objects are the essential tools of relational database. These database objects helps to store, view, edit and manipulate the data and information stored in database.

It can be used to hold and manipulate the data. Some of the examples of database objects are view, sequence, indexes, form, query report etc.

- **Table:** Basic unit of storage; composed rows and columns
- **View:** Logically represents subsets of data from one or more tables
- **Sequence:** Generates primary key values
- **Index:** Improves the performance of some queries
- **Synonym:** Alternative name for an object

Some Basic Terms used in Database

- ***Schema:*** A schema is the structure of database which defines name of tables, data fields with data types, relationships and constraints.
- ***Instance:*** It defines data values in a record.
- ***Entity:*** An entity is a thing or object in the real world that is different from other objects.
- ***Attribute:*** Attribute is properties possessed by an entity or relationship.
- ***Index:*** It is used to create indexes in database. It helps searching and sorting operation faster and improves the performances of queries.
- ***Query:*** It is the object of DBMS which is mainly used to extract and upgrade the necessary records that are present in the database.
- ***Form:*** It is object of database which is mainly used for data entry. It is easy to add, modify and delete the records in form.
- ***Report:*** Report are the printed output that is created from table or query. We can't add, modify and delete the records in report.

Keys of DBMS:

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.

- **Primary Key:** A primary key is one or more columns in a table used to uniquely identify each row in the table. Primary key cannot contain Null value.
  A primary key is a special relational database table column (or combination of columns) designated to uniquely identify each table record. A table cannot have more than one primary key.
  A primary key's main features are:
  - It must contain a unique value under the field.
  - It cannot contain null values.
  - Every row must have a primary key value.
- **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.
- **Candidate Key:** If a relational schema has more than one key, that 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.
- **Alternate Key/ Secondary Key:** Alternative 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 alternative.
- **Compound Key:** It has two or more attributes that allow you to uniquely recognize specific record. It is possible that each column may not be unique by itself within the database.

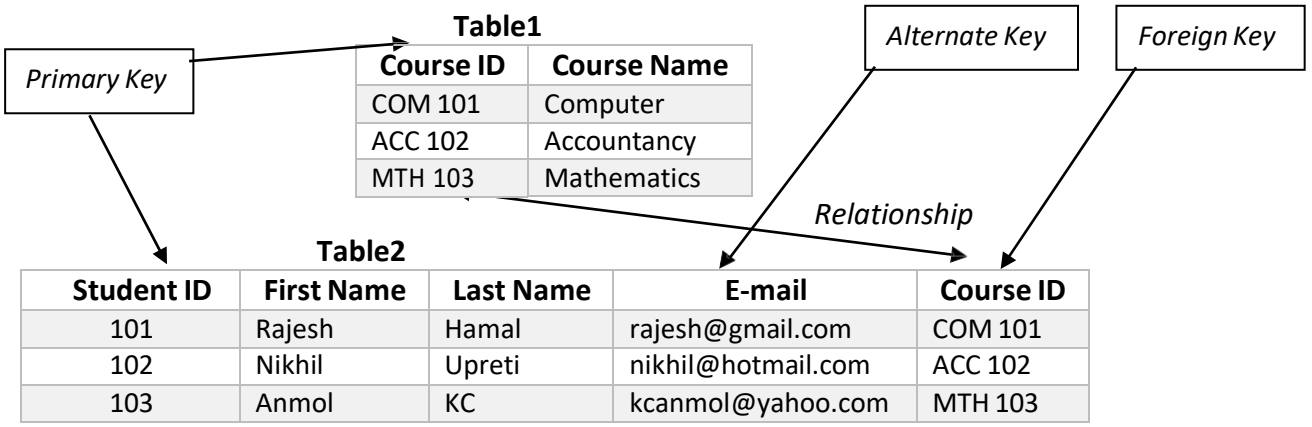


Table1

Course ID	Course Name
COM 101	Computer
ACC 102	Accountancy
MTH 103	Mathematics

Table2

Student ID	First Name	Last Name	E-mail	Course ID
101	Rajesh	Hamal	rajesh@gmail.com	COM 101
102	Nikhil	Upreti	nikhil@hotmail.com	ACC 102
103	Anmol	KC	kcanmol@yahoo.com	MTH 103

SQL (Structured Query Language):

It is an international standard database query language for accessing and managing data in the database.

Features of SQL

- It is a non-procedural Language.
- It is English like language.
- It can process a single record as well as sets of records at a time.
- It is a data sub-language consisting of three built in language: DDL, DML, DCL etc.
- It insulates the user from the underlying structure and algorithm.
- It has the facilities for defining tables, views, security, integrity, transaction control etc.

1. **DDL (Data Definition Language):** DDL is used by the database designers and programmers to specify the content and structure of the table. It is used to define the physical characteristics of records. It includes commands that manipulate the structure of objects such as *views, tables, and indexes*, etc.

» CREATE: Create is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
» DROP: Drop is used to delete objects from the database.
» ALTER: Alter is used to alter the structure of the database.
» TRUNCATE: Truncate is used to remove all records from a table, including all spaces allocated for the records are removed.
» COMMENT: Comment is used to add comments to the data dictionary.
» RENAME: Rename is used to rename an object existing in the database.

2. **DML (Data Manipulation Language):** DML is related with manipulation of records such as retrieval, sorting, display and deletion of records of data. It helps user to use query and display reports of the table. So it provides technique for processing the database.

» SELECT: It is used to retrieve data from a database.
» INSERT: It is used to insert data into a table.
» UPDATE: it is used to update an existing data in table.
» DELETE: It is used to delete record from table.

3. **DCL (Data Control Language):** DCL provides additional features for security of table or database. It includes commands for controlling data and access to the database. Examples of these commands are *commit, Rollback, Grant* etc.

➢ **GRAND:** It gives user's access privileges to database.
➢ **REVOKE:** It is used withdraw users' access privileges given by using the GRANT command.

Database Model:

A Database model defines the logical design and structure of a database and are used to show how data will be stored, accessed and updated in a Database Management System. It refers to the layout of a database and helps in designing a database. The various types of database models are

Different database models

1. ***Hierarchical database model:*** this is one of the oldest types of database models. In this model data is represented in the form of records. Each record has multiple fields. All records are arranged in database as tree like structure. The relationship between the records is called parent child relationship in which any child record relates to only a single parent type record.

Advantages

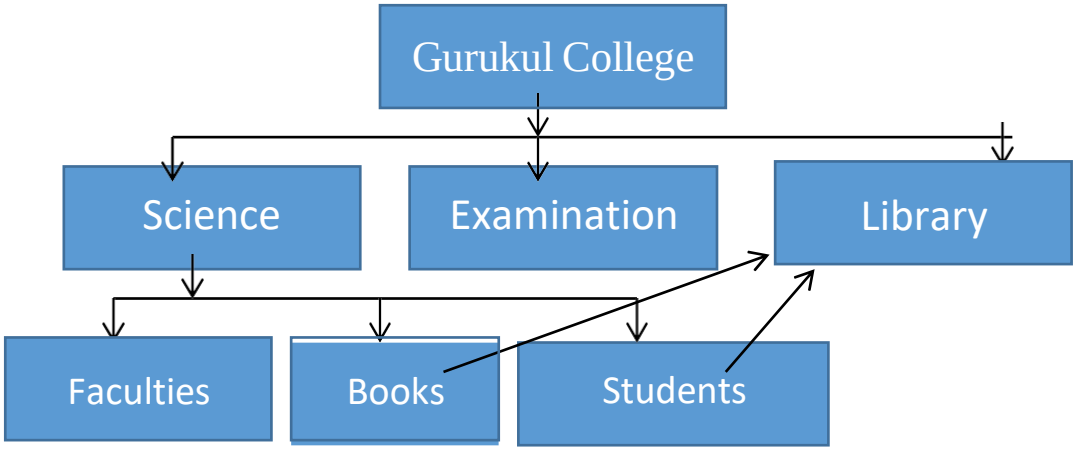
- It is the easiest model.
- Searching is fast and easy if parent is known.

- It is very efficient in handling one to one and one to many relationships.

Disadvantages

- It is old and outdated database model.
- It does not support many to many relationships.
- It increases redundancy because same data is to be repeated in different places.

2. ***Network database model:*** It replaced hierarchical network database model due to some limitations on the model. Suppose, if an employee relates to two departments, then the hierarchical database model cannot able to arrange records in proper place. So network, database model was emerged to arranged non-hierarchical database. *The structure of database is more like graph rather than tree structure. A network database model is a database model that allows multiple records to be linked the same owner file. The network model allows each child to have multiple parents.*



Advantages

- It accepts many to many relationships, so it is more flexible.
- The searching is faster because of multidirectional pointer.
- The network model is simple and easy to design.
- It reduces the redundancy.

Disadvantages

- It is difficult to handle the relationship in complex programs.
- There is less security because of sharing data.
- It increases the processing overhead due to the complex relationship.

3. ***Relational database model:*** in this model, the data is organized into tables which contain multiple rows and columns. These tables are called relations. A row in a table represents a relationship among a set of values. Since a table is a collection of such relationships, it is generally referred to the mathematical term relation, from which the relational database model derives its name.

We notice from below table, here each student has a unique roll number and has marks of subject. Here Roll makes relation between these two tables.

Table: Student

Roll	Name
1	Hari
2	Sita
3	Ram

Table: Subject

Roll	Math	English	Computer
1	80	90	95
2	90	80	85
3	95	95	95

For example, if we make relation between student and subject, we get

following:

Table: Subject				
Roll	Name	Math	English	Computer
1	Hari	80	90	95
2	Sita	90	80	85
3	Ram	95	95	95

Advantages

- The breaking of complex database table into simple database table becomes possible.
- Database processing is faster than other model.
- There is very less redundancy.
- The integrity rules can easily be implemented.

Disadvantages

- It is more complex than other models.
- There are too many rules because of complex relationships.
- It needs more powerful computers and data storage devices.

4. ***Object oriented database 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.

5. Entity Relationship Diagram:

The diagrammatic representation of entities attributes and relationship is called E-R diagram. The E-R diagram is an overall logical structure of a database that can be expressed graphically. It was developed to facilitate database design. It is graphical representation of database.

Components of E-R Diagram

Entity	Attribute	Link	Relationship

- **Entity:** An entity is defined as anything about which data to be collected and stored.
- **Relationships:** Relationships describes associations among data. Most relationships describes associations between two entities.
- **Attribute:** Attribute describes particular characteristics of the entity.

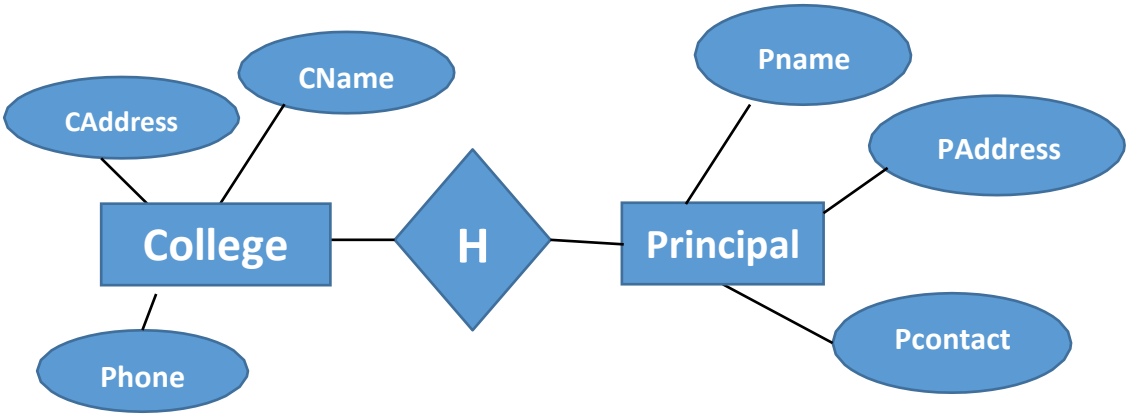


Fig. E-R diagram for relation between college and principal

Relationship and its types:

A relationship is an association among several entities and represents meaningful dependencies between them. It is represented by diamond. There are 3 types of relationship:

i. One to one
ii. One to many
iii. Many to many

1. One to one: if one record of an entity is related with only one record of another entity then such type of relationship is called one to one relationship.

- College---------- Principal
- Bank----------Manager
- Driver ---------- Car

2. One to many: If one instance of one entity is related with many instances of other entity then it is called the one to many relationship.

- College----------students
- Bank----------Employers

3. Many to many: If many instances of the one entity are related with many instances of another entity then it is called many to many relationship.

- Teachers---------- students
- Books ---------- Readers
- Employers ---------- Customers

Concept of Normalization

Normalization is a database design process in which complex database table is broken down into simple separate tables. It makes data model more flexible and easier to maintain.

- Database Normalization is a technique of organizing the data in the database. It is a systematic approach of decomposing the tables to eliminate data redundancy and inconsistency. The data is said to be redundant if there is duplicate or repeated data in the table.
- Normalization divides the larger table into the smaller table and links them using relationship. It increase clarity in organizing data in the database.

For example:

Below table shown is our database without normalized. Here in table we can see that for the large records of this table, there would be multiple data row of same values especially in the country and city column. So, we can normalize the table by splitting it into two tables where one table only stores the location area of each person name and could be referenced by some unique id. Say Area code.

Id	Country	City	Name
101	Nepal	Kathmandu	Alex
102	India	Delhi	Martin
103	Nepal	Kathmandu	Melman
104	Japan	Tokyo	Gloria

The above table can be normalized in two tables as below:

country	City	Area code
Nepal	Kathmandu	N1
India	Delhi	I1
Japan	Tokyo	J1

Id	Area code	Name
101	N1	Alex
102	I1	Martin
103	N1	Melman
104	J1	Gloria

Advantages of Normalization

- It reduces the data redundancy.
- It improves faster sorting and index creation.
- It improves the performances of the database system.
- It simplifies the structures of table.
- It avoids the loss of information.

Disadvantages of Normalization

- It is complex to design due to the relationship between tables.
- It requires more CPU cycles.
- It requires large memory.
- It requires more joins to get the result.
- Maintenance overhead.

Types of Normalization

❖ **1NF (First Normal Form):**

A table is said to be in first normal form if it has atomic values. There shouldn't be any repeating groups of attribute in the table. First normal form sets the very basic rules for an organized database.

- The data field should be a single (atomic) valued attribute/ columns.
- It eliminates duplicates rows and columns from the same table.
- It minimizes the data redundancy in the database table.

❖ **2NF (Second Normal Form):**

A table is said to be in 2NF if it is a First normal form and it doesn't have the partial dependency. Second normal form further addresses the concept of removing duplicate data.

- It should be in the first normal form.
- It should not have partial Dependency.
- It identifies data dependencies.
- Non key attributes are functionally depends on key attribute (primary key).

❖ **3NF (Third Normal Form):**

Third normal form goes one large step further.

- It should be in the second normal form.
- It removes transitive dependencies in a table.
- All non-primary key attribute must dependent on primary key attribute or attribute.

In Details Normalization with Examples

Un-normalized:

A table is said to be un-normalized when there is repetition of data in a table. In un-normalized table records are not atomic. Let's take an example of un-normalized table.

Un-normalized table: **Table No.1**

Roll No.	Name	Faculty	Subject
1	Sundar	ICT	Java, OS
2	Mukesh	ICT	Network
3	Ganesh	ICT	C, Web

A. 1NF (First Normal Form):

A table is said to be in first normal form if it has atomic values. There shouldn't be any repeating groups of attribute in the table. Following are the main rules for table to be in 1NF:

- Table should have single (atomic) valued attributes/columns.
- Values stored in columns should be of same domain.
- Columns name should not be repeated in table.
- The order of column names doesn't matter.

The table given above in un-normalized data meets the three requirements among four to be in first normal form. In the subject column more than one subject are stored in a single column for two students. But, each column must contain atomic value to be in first normal form. And the problem is solved in the table given below:

Example of 1NF for above table No.1

Table No.2

Roll No.	Name	Faculty	Subject
1	Sundar	ICT	Java
1	Sundar	ICT	OS
2	Mukesh	ICT	Network
3	Ganesh	ICT	Web
3	Ganesh	ICT	C

Though, some values are repeated but all columns are atomic for each record /row.

B. 2NF (Second Normal Form):

A table is said to be in 2NF if it is in First normal form and it doesn't have the partial dependency. i.e. each attributes should functionally depend on primary key. Rules for 2 NF:

- A table should be in first normal form.
- There must not be partial dependency.
- Partial dependency exists when any attribute of a table depends on only one part of a composite primary key (primary key combining more than one field) and not on the complete primary key.
- To remove partial dependency, a table can be divided and attributes creating partial dependency are removed in some other tables.

Situation of Dependency:

Let's take an example of table student with student_id, name, address and age as its columns.

Student_id	***Name***	***Address***	***age***

Here student_id is the primary key which can identify each records uniquely and can be used to fetch any row of data from this table.

Student_id	Name	Address	Age
15	Ganesh	KTM	17
16	Janaki	BKT	17

Here we can get name, address and age of the student easily from their student_id. Which means each column can be fetched using primary key. So, all needed is student_id and every other column depends on it or can be fetched using it.

This is called dependency or functional dependency. And this kind of dependency must be in table to be in second normal form.

Situation for partial dependency:

In above table a single filed student_id uniquely identifies the all the records of the table. But in some cases combination of two or more columns or fields makes the primary key. Where more than one field acts as primary key. Lets create a table named subject with fields subject_id and subjectname.

Subject_id	Subjectname
101	Math
102	Science
103	Nepali

Above we have two tables: student and subject for storing student's and subject's information. Now, let's make a table named **Mark** storing student's mark in respective subjects with subject teacher.

Score_id	Student_id	Subject_id	Marks	Teacher
1	15	101	55	Bishnu
2	15	102	65	Umesh
3	16	103	88	Janvi

Note: the above table is not in 2nd normal form.

In above table student_id is used to get student's information where as subject_id is used to get subject name. The combination of student_id and subject_id is the primary in above table. It is because if we want to get mark of student with id 15 then we cannot get because we don't know which subject. Here we have to give sudent_id and subject_id to uniquely identify any row.

Is there a partial dependency in above table? Obviously, yes. In the given table **Mark** column name teacher is only dependent on the subject, for math there is Bishnu for science Umesh and so on. But the primary key is the combination of student_id and subject_id, teacher's name depend only on subject, i.e. subject_id not on the student id.

This situation is known as **partial dependency**, where an attribute/ column in table depends on only a

part of primary key not on the whole key.

Removing the partial dependency:

Above table can be normalized in second normal form by removing teacher's name from the **Mark** table adding it to **Subject table.**

Subject:

Subject_id	Subjectname	Teacher
101	Math	Bishnu
102	Science	Umesh
103	Nepali	Janvi

Mark:

Score_id	Student_id	Subject_id	Marks
1	15	101	55
2	15	102	65
3	16	103	88

Summary:

- For table to be in second normal form, it should be in first normal form.
- Partial dependency exists, when non primary key attribute depends only on a part of primary key instead of complete primary key.
- Partial dependency can be removed by breaking a table and removing attributes causing partial dependency.

C. 3NF (Third Normal Form):

A table is said to be in third normal form, if it is second normal form and it doesn't have any transitive dependency in primary key. The elements that are not dependent on primary key are removed. Transitive dependency occurs in table when a non-primary key attribute depends upon another non primary key attribute. All non-primary key attribute must dependent on primary key attribute or attributes.

Transitive Dependency?

Transitive Dependency occurs when a non-primary key attribute depends upon another non primary key attribute instead of primary key attribute or primary key.

For instance:

In the above table **Mark**, lets add some more information such as Exam_name and Full_mark.

Score_id	Student_id	Subject_id	marks	Exam_name	Full_marks
1	15	101	55	First Term	500
2	15	102	65	First Term	500
3	16	103	88	Second Term	300

In above table, student_id and subject_id are the primary key. The column exam_name depends on both student_id and subject_id. But, the Full_marks depends on the Exam_name. The first term exam might have 500 full mark but the second term may have 300 or others. Here exam_name is neither primary key nor the part of primary key still, Full_mark depends on it. So, here full_mark which is non-primary key attribute depends on another nonprimary key attribute known as Exam_name. This situation is known as **transitive dependency**.

Removing Transitive Dependency:

Again, table should be broken into small individual tables to remove it. So we need to remove those fields which are creating transitive dependency. Which looks like.

Score Table:

Score_Id	Student_Id	Subject_Id	Marks	Exam_Id
1	15	101	55	11
2	15	102	65	12
3	16	103	88	13

Exam Table:

Exam_ID	Exam_Name	Full_Marks
11	First Term	500
12	Second Term	300

Benefits of removing transitive dependency:

- Amount of data duplication is reduced.
- Data integrity is achieved.

Note: Normalization does not eliminate data redundancy. Instead, it reduces the redundancy.

Example of normalization:

Un-normalized Table:

Employee Id	Name	Address	Department
101	Ram	Kathmandu	Sales
102	Bikky	Bhaktapur	Marketing, Export
103	Anusha	Lalitpur	import

First Normal Form:

Employee Id	Name	Address	Department
101	Ram	Kathmandu	Sales
102	Bikky	Bhaktapur	Marketing
102	Bikky	Bhaktapur	Export
103	Anusha	Lalitpur	import

Second Normal Form:

Lets take a table employee having more than one department.

Employee Id	Department	Salary
101	Sales	20000
102	Marketing	25000
102	Export	25000
103	import	20000

Here the non-primary key attribute salary dependent on the employee id only. Here Employee id and department are the candidate key. This violates the rule that “no non primary attribute is dependent on the part of primary key or on the subset of candidate key.”

To make table in 2NF we can break it as:

Employee Id	Salary
101	20000
102	25000
103	20000

Employee Id	Department
101	Sales
102	Marketing
102	Export
103	import

Third Normal Form:

Employee Id	Name	Department	HoD
101	Rikesh	HR	Mukesh
102	Binita	Marketing	Mukesh
103	Jagdish	Store	Bikash

Here, Employee Id is the primary key and all other are non-primary key attributes. The non- primary key attribute HoD is dependent on non-primary key attribute Department. Here, transitive dependency occurs. To remove it we can decompose table as:

Employee Id	Department Id	Name
101	A10	Rikesh
102	A11	Binita
103	A12	Jagdish

Department ID	Department	HoD
A10	HR	Mukesh
A11	Marketing	Mukesh
A12	Store	Bikash

Emp_code	January
Emp_name	February
Address	March
Contact no.	April
Date of birth	May
Department	June
Designation	July
Basic_salary	Daily_allowance
Travel_Allowance	Gross_salary
Tax	Provident _fund

Normalized database

Employee	Salary	Month
Emp_code	Basic_Salary	January
Emp_name	Travel_allowance	February
Address	Daily_allowance	March
Contact no.	Gross_salary	April
Date of birth	Provident_funt	May
Department	Tax	June
Designation		July

Example 2

Name	Roll	Class	Sub_name	Sub_marks	Sub_name	Sub_marks
Ram	1	11	Computer	95	Account	78
Sita	1	12	Computer	98	Account	80
Hari	2	11	Computer	80	Account	82
Shyam	2	12	Computer	92	Account	83

In above table, we can see that column of subject nome and marks are repeated which are eliminated in **1NF.**

Name	Roll	Class	Sub_name	Sub_marks
Ram	1	11	Computer	95
Ram	1	11	Account	78
Sita	1	12	Computer	98
Sita	1	12	Account	80
Hari	2	11	Computer	80
Hari	2	11	Account	82
Shyam	2	12	Computer	92
Shyam	2	13	Account	83

In above table name depends upon roll no and class, subject name only depends upon class, subject marks depends upon name and subject_name. Hence, above table can be decomposed as **2NF:**

Subject	Class
Computer	11
Account	11
Computer	12
Account	12

Name	Roll	Class
Ram	1	11
Sita	1	12
Hari	2	11
Shyam	2	12

Name	Sub_name	Sub_marks
Ram	Computer	95
Ram	Account	78
Sita	Computer	98
Sita	Account	80
Hari	Computer	80
Hari	Account	82
Shyam	Computer	92
Shyam	Account	83

It removes the column that are not dependent on primary key using **3NF** above table can be decomposed as:

Sub_id	Subject
C1	Computer
A1	Account

Class_id	Class
XI	11
XII	12

Std_id	Class	Roll	Class_id
1	Ram	1	XI
2	Sita	1	XII
3	Hari	2	XI
4	Shyam	2	XII

Std_id	Sub_id	Marks
1	C1	95
1	A1	78
2	C1	98
2	A1	80
3	C1	80
3	A1	82
4	C1	92
4	A1	83

Centralized database system Vs. Distributed database system:

Centralized database system:

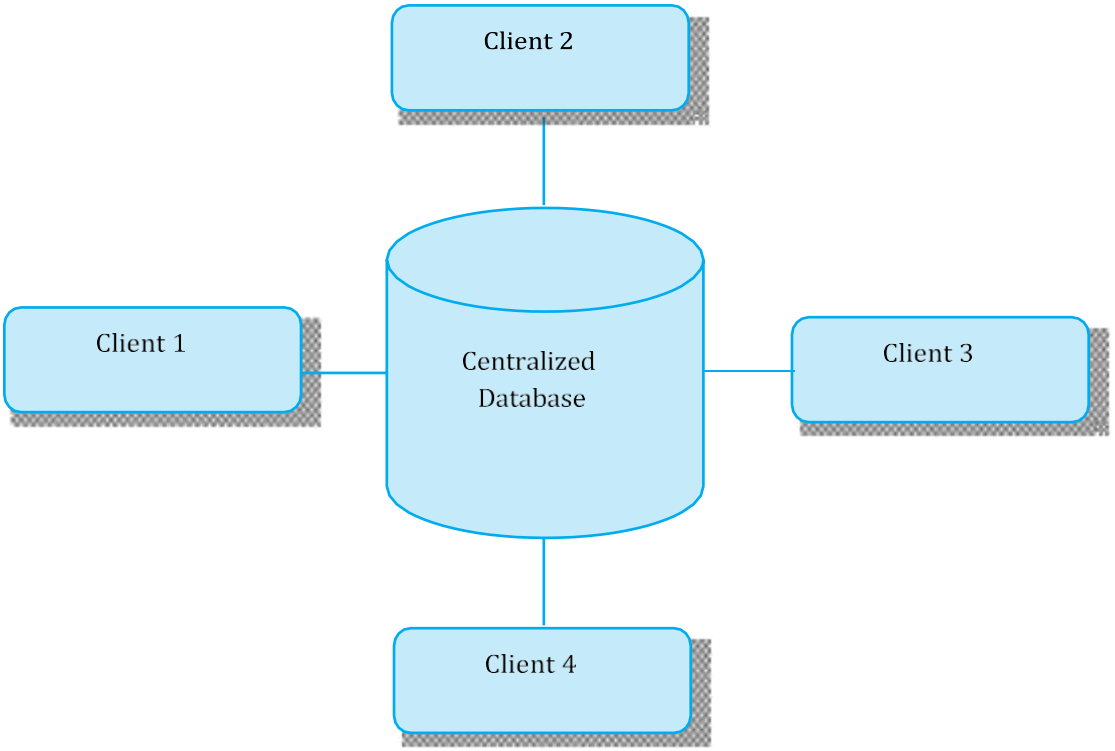
- *The database system where data and information are stored in the centralized server or centralized database system.*
- The data stored in database are accessed from different locations through several applications. The information (data) is stored at a centralized location and the users from different locations can access this data.
- This type of database contains application procedures that help the users to access the data even from a remote location.

Advantages:

- It decreases risk of data manipulation. i.e. manipulation of data will not affect the core data.
- Data consistency is maintained as it manages data in a central repository.
- It provides better data quality, which enables organizations to establish data standards.
- It is less costly as fewer vendors are required to handle the data sets.

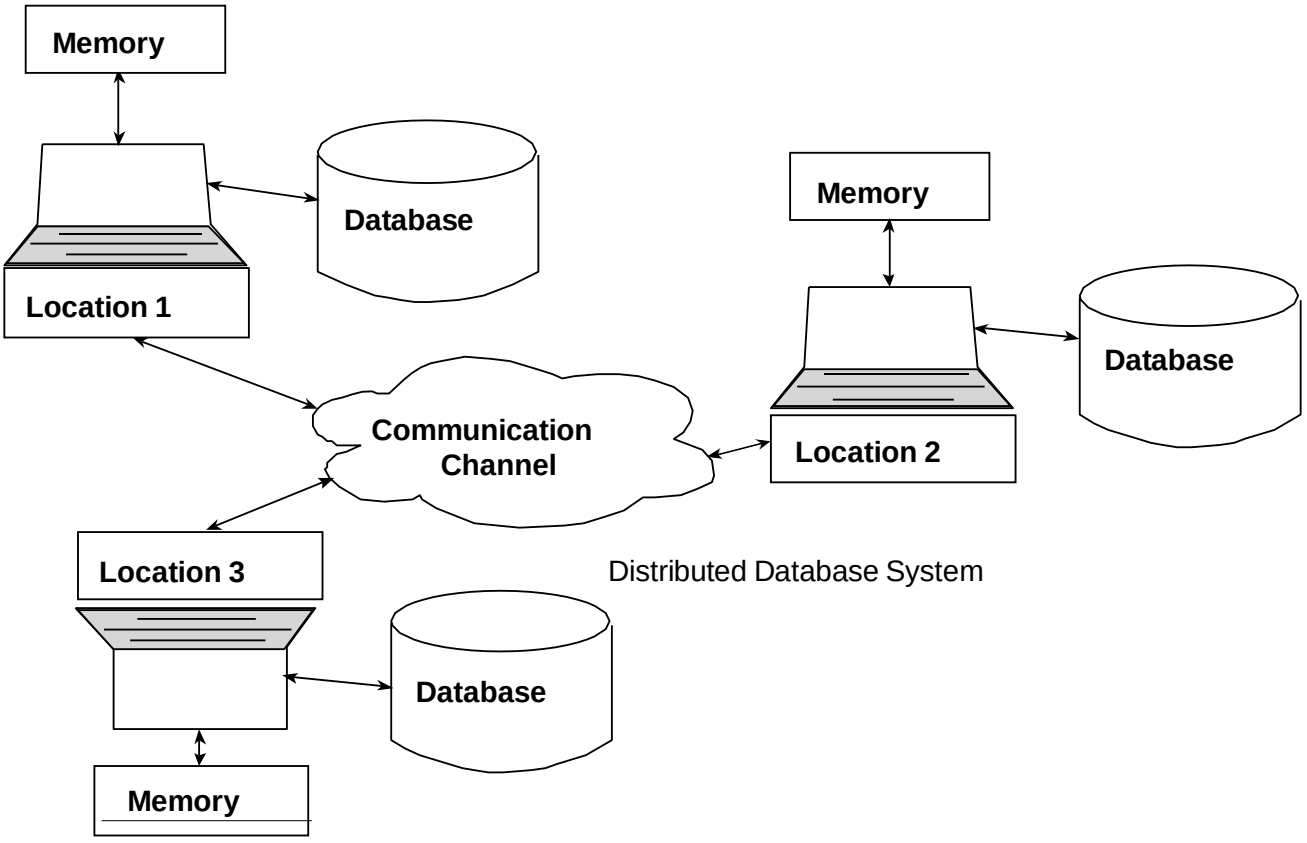
Disadvantages:

- The size of centralized database is large which increases the response time of fetching data.
- It is difficult to update the centralized database.
- If sever gets damaged entire data will be lost.



Distributed database system:

- Distributed database doesn't store all data and information in the single but store on various sites or places, which are connected by the help of communication, links which helps them to access the distributed data easily.
- In distributed database various portions of a database are stored in multiple different locations along with the application procedures which are replicated and distributed among various points in a network.



Distributed Database System

Advantages:

- The system can be expanded by including new computers and connecting them to the distributed system.
- Distributed database is more reliable than centralized database.
- The performance and service are better.
- Large numbers of users are supported.
- One server failure will not affect the entire data set.

Disadvantages:

- It is difficult to administrate and manage the database
- It is expensive to set up.
- This database has high risk of hacking and data theft.

Different between centralize and distributed database system

Centralized database system	Distributed database system
1. Simple type	1. Complex type
2. Located on particular location	2. Located in many geographical locations.
3. Consists of only one server	3. Contains servers in several locations
4. Suitable for small organizations	4. Suitable for large organizations
5. Less chance of data lost	5. More chances of data hacking, lost
6. Maintenance is easy and security is high	6. Maintenance is not easy and security is low
7. Failure of system makes whole system down	7. Failure of one server does not make the whole system down
8. There is no feature of load balancing	8. There is feature of load balancing
9. Data traffic rate is high	9. Data traffic rate is low
10. Cost of centralized database system is low	10. Cost of distributed database system is high

Data dictionary:

A data dictionary is a file which contains meta-data that is data about data. It also called information system catalogue. It keeps all the data information about the database system such as location, size of the database, tables, records, fields, user information, recovery system, etc.

Data integrity:

Data integrity referees to the validity or consistency of data in database. It ensures that the data should be accurate and consistent.

Mainly there are 3 types of data integrity constraints used in the database system. They are as:

1. ***Domain integrity constraints:*** it defines a set range of data values for given specific data field. And also determines whether null values are allowed or not in the data field.
2. ***Entity integrity constraints:*** it specify that all rows in a table have a unique identifier, known as the primary key value and it never be null i.e. blank.

3. ***Referential integrity constrains:*** it exists in a relationship between the two tables in a database. It ensures that the relationship between the primary keys in the master table and foreign key in child table are always maintained.

Data Security:

Data security is protection of data in database system against unauthorized access, modification, failure, losses or destruction. The authorized access means only right people can get the right access to the right data.

DBA (Database Administrator)

DBA is the most responsible person in an organization with sound knowledge of DBMS. He/she is the overall administrator of the program. He/she has the maximum amount of privileges for accessing database and defining the role of the employee which use the system. The main goal of DBA is to keep the database server up to date, secure and provide information to the user on demand.

Qualities of good DBA

1. He/she should have sound and complete knowledge about DBMS and its operation.
2. He/she should be familiar with several DBMS packages such as MS Access, MY SQL, Oracle etc
3. He/she should have depth knowledge about the OS in which database server is running.
4. He/she should have good understanding of network architecture.
5. He/she should have good database designing skill.

Responsibilities

1. DBA has responsibility to install, monitor, and upgrade database server.
2. He/she should has responsibility to maintain database security by creating backup for recovery.
3. He/she has responsibility to conduct training on the uses of database.
4. DBA defines user privilege, relationships and manages form, reports in database.

The SQL statement for creating, dropping, and altering database and table

XAMPP provides a GUI environment to perform any operations on the database. However, it also provides an option to use SQL statements to perform any operations SQL statements are used in the SQL menu in phpMyAdmin. The SQL statements used in XAMPP also work well with most of the databases.

Creating a database:

Syntax: CREATE DATABASE databasename;

Example: CREATE DATABASE School;

Dropping the database (deleting the database):

Syntax DROP DATABASE databasename:

Example: DROP DATABASE School:

Creating a table:

Syntax CREATE TABLE table_name (column1 datatype, column2 datatype, column3 datatype….)

Example: CREATE TABLE Students (StudentID int, FName varchar(255), LName varchar(255), Address varchar(255), Class varchar(255));

Altering table adding, deleting, or modifying columns in an existing table

Adding column Syntax:

Synatx:ALTER TABLE table name

ADD column_name datatype

Example: ALTER TABLE Students ADD Email varchar(255));

Deleting column

Syntax:

ALTER TABLE table name

DROP COLUMN column_name;

Example: ALTER TABLE Students

DROP COLUMN Email

Modifying column (changing the data type of a column in a table)

Sytnax:

ALTER TABLE table name MODIFY COLUMN column_name datatype:

Example: ALTER TABLE Students

MODIFY COLUMN Class int

Deleting table

DROP TABLE table_name,

SQL statement to insert, select, update and delete data

Syntax Example: DROP TABLE Students:

Inserting data

Syntax

INSERT INTO table name (column1, column2 column3,...) Example: INSERT INTO Students (StudentID, FName, LName, Address, Class)

VALUES (valuel, value2, value3.);

VALUES ('101', 'Ram', Sharma', Pokhara', 7);

Selecting data iselecting data from a database);

Syntax SELECT column1, FROM table name column2.

Example: SELECT * FROM Students; (This will select all the columns from the table Students] SELECT FNAME. LNAME FROM Students: [This will select only the First Name and Last Name from the table Students.]

Selecting data using conditions

Syntax SELECT column1, column2....

FROM table_name

WHERE condition;

Web References:

- https://www.javatpoint.com
- https://www.w3schools.com
- https://www.tutorialspoint.com
- https://www.google.com
- https://www.wikipedia.org

COMPUTER SCIENCE

Grade: XII

DATA COMMUNICATION AND NETWORKING

REFERENCE NOTE

NEB Important Questions for Computer Science XII

Unit-2- Data Communication and Networking **- 9 Marks**

1. ***What is Computer Network? Explain advantages and disadvantages of computer network.***
2. Discuss different types of computer network on the basis of size.
3. ***What do you mean by network topology? Explain the different types of network topologies.***
4. ***Differentiate between LAN and WAN.***
5. ***Define network architecture. Differentiate between client server and peer to peer networks.***
6. Differentiate between Simplex, Half Duplex and Full Duplex.
7. Differentiate between internet and intranet.
8. What is Internet addressing? Discuss Network Tool.
9. ***What do you mean by transmission media? Explain the different types of transmission media used in computer network?***
10. ***What do you mean by OSI reference model***? Explain the different layers of OSI reference model.
11. Define the following terms : a) Repeater ***b) Bridge*** c) Protocol ***d) Router*** e***) Satellite*** ***f) Microwave System*** ***g) subnet Mask*** ***h) MAC address*** ***i) Coaxial Cable*** ***j) Fiber Optic Cable***

Unit 2- Data Communication and Networking

Concept of Communication System

The exchange of data and information between sender and receiver through any given medium following a common rule is known as communication. So, telecommunication is the art of communicating at a distance. Telephone, radio and television are the main and popular media of tele-communication.

Data Transmission

Data may be transferred from one device to another by means of some communication media. The electromagnetic or light waves that transfer data from one device to another device in encoded form are called **signals**. Data transmissions across the network can occur in two forms i.e.:

a. Analog signal

b. Digital signal

Analog Signal

The transfer of data in the form of electrical signals or continuous waves is called analog signal or analog data transmission. An analog signal is measured in volts and its frequency is in hertz (Hz).

Digital Signal

The transfer of data in the form of digit is called digital signal or digital data transmission. Digital signals consist of binary digits 0 & 1. Electrical pulses are used to represent binary digits. Data transmission between computers is in the form of digital signals.

Modulation

The modulation is the process of changing or encoding the carrier wave at certain amplitude (height) and frequency (timing) is called modulation.

The process of changing some characteristics (amplitude, frequency or phase) of a carrier wave in accordance with the intensity of the signal is known as modulation. Modulation means to change.

There are three types of modulation:

a. ***Amplitude Modulation (AM):*** Amplitude modulation is an increase or decrease of the carrier voltage (Ec), will all other factors remaining constant.

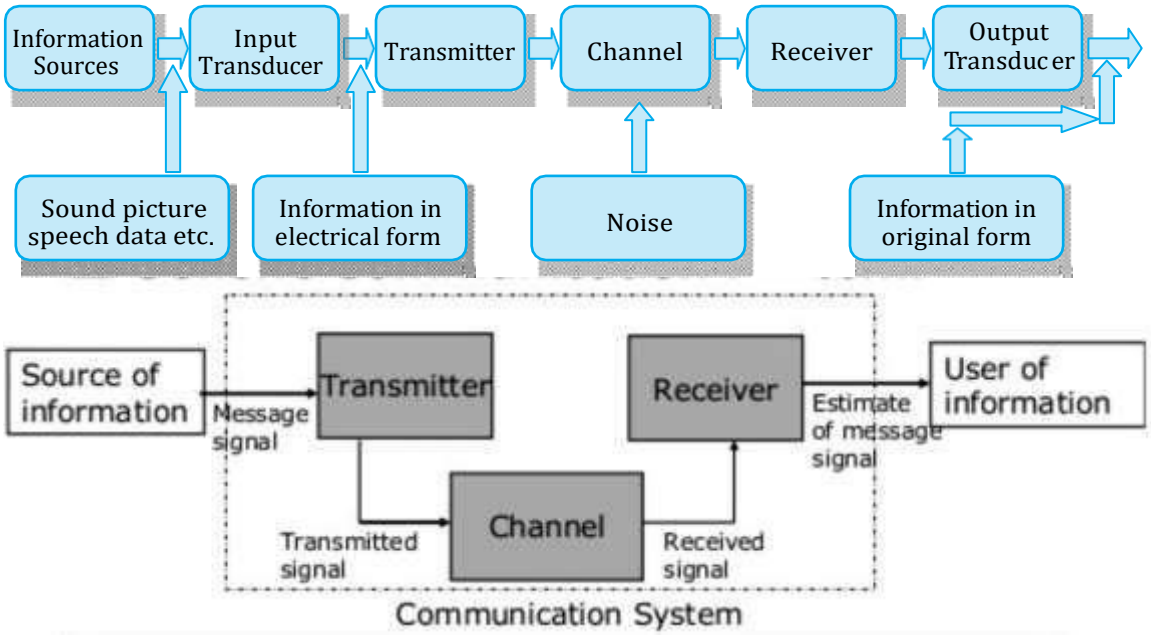
b. ***Frequency Modulation (FM):*** Frequency modulation is a change in the carrier frequency (fc) with all other factors remaining constant.

c. ***Phase Modulation (PM):*** Phase modulation is a change in the carrier phase angle (Î¸). The phase angle cannot change without also affecting a change in frequency. Therefore, phase modulation is in reality a second form of frequency modulation.

Block diagram of Communication System

Exchanging of data, information and message is known as communication in simple term. The devices, equipment and other necessary things which play vital role in the communication of message form source to destination are known as communication equipment. These equipment may be computer, mobile, fax and other devices. When all these communication equipment combine together for exchanging the message, then they form the communication system. Line telephony, radio, television broadcast, mobile communication, computer communication etc. are the common examples of communication system. In the communication, initially the message, data are originated from the information source and sent to

destination passing several stages. The block diagram of general communication system is described below:



Elements of Communication System

a. Information Source:

Information source is the place or device which generates the information in communication system. The information may be in different forms such as text, data, speech, pictures, words etc. these information are send to the destination through the communication system. Main role of Information source is to generate the required message which has to be transmitted.

b. Input Transducer:

Input Transducer is an important device which convert one form of energy into another form. In some cases, the message generated form the information source may not be electrical in nature. At that time input transducer is responsible for converting such messages into electrical form. For instance, microphone converts analog sound in electrical signals.

C. Transmitter:

As the name suggests transmitter is responsible for transmitting and processing the electrical signals through the communication channel. In long distance radio communication, signal amplification is necessary for transferring data. Transmitter processes the signals such as restriction of range of audio frequencies, amplification and modulation of signals. All these processes are done just to ease the transmission of signals through the channel.

D. Transmission channel or channel:

Transmission channel is the transmission medium through data signals or information passes from source to destination. It is the medium through which the message travels from the transmitter to the receiver. It provides the connection between the sender / transmitter or receiver. Communication channel may be wired/ guided or wireless/unguided. The wired medium or also called point to point channels where devices are connected through the cables. Whereas wireless medium connect devices wirelessly through electromagnetic wave signals.

Sometime unwanted signal tend to interfere with the information that unwanted signal is called noise.

E. Receiver:

The main function of the receiver is to reproduce the message signal in electrical form from the distorted received signal. This reproduction of the original signal is accomplished by a process known as the demodulation or detection. Demodulation is the reverse process of modulation carried out in transmitter.

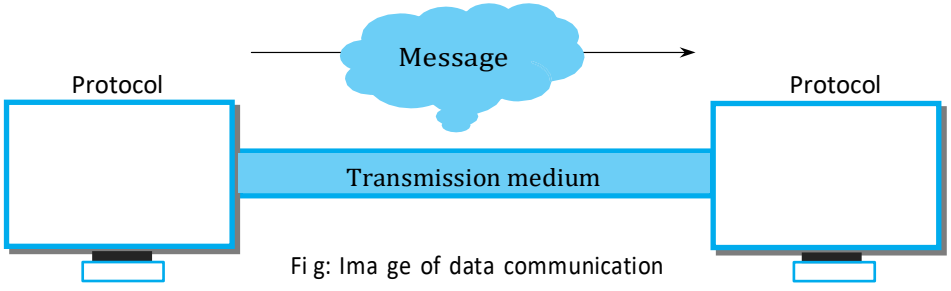
F. Output Transducer/Destination:

The transducer presented at output side of the communication system is called output transducer. It is responsible of converting electrical signals into non electrical signals like sound. It is also known as destination and is responsible for converting message signals into its original form from the electrical form. For instance, speaker is a destination or transducer in voice communication.

Elements of Data communication System

Data communication is the process of exchanging data among the computing devices. Data may be in different form like file, text, image, sound etc. These data are transmitted between a source and a destination. Source device is responsible for generating data and destination device is responsible for receiving the data generated by source. Communication like, email, IRC, VOIP etc. are examples of data communication. Data communication is the transmission of electronic from one computing device to another. The data communication comprises of following elements:

1. **Sender (source)**: It is the device that generates and sends that message.
2. **Medium:** It is the channel or physical path through which the message is carried from the sender to the receiver. The medium can be wired like twisted pair wire, coaxial cable, fiber-optic cable or wireless like laser, radio waves, and microwaves.
3. **Receiver (sink)**: It is the device that receives the message.
4. **Data:** It is the information or data to be communicated. It can consist of text, numbers, pictures, sound or video or any combination of these.
5. **Protocol:** It is a set of rules that govern the communication between the devices. Both the sender and the receiver follow same protocols to communicate with each other.



Fi g: Ima ge of data communication

The data communication takes place between two devices connected together either by wired or wireless media. The sender device generates the data message and send it through the communication channel using some protocol. After the message is sent the receiver receives the message using the similar protocol.

6. **Encoder:** The computer works with digital signals. The communicational channels usually use analog signals. Therefore, to send data through a communication channel, the digital signals are encoded into analog signals or into a form which can be transmitted through transmission medium. This is called encoding. The device that carries out this function is called encoder.

7. **Decoder:** The Computer works with digital signals. The communication channels usually use analog signals. Therefore, to receive data from a communication channel, the coded analog signals or any other encoded form is converted back to digital signals. This is called decoding. The device that carries out this function is called decoder.

Mode of Communication

Mode of communication normally refers to the ways in which communication takes place or data gets transmitted between source and destination nodes. Modes of data transmission direct the direction, mechanism of data flow between devices.

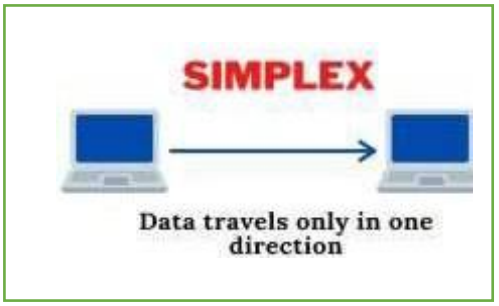
1. **Simplex Communication:**

In this type of communication data transmission takes place only in one direction. It is also called a unidirectional communication mode.

Radio, Television, Newspaper and keyboard to CPU Communication are some of the most common example of simplex communication.

Features:

- Data are transmitted in only one direction.
- The sender sends data and receiver receives only.
- There is not bidirectional communication.
- Listeners cannot reply immediately.
- Radio and television broadcast are its examples.



2. **Duplex Communication:**

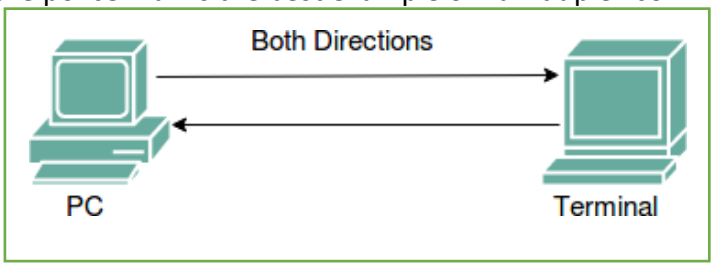
In duplex communication, mode data transmission is possible from both directions. The receiver can immediately respond to the sender. The duplex communication can be categorized into two groups.

a. Half Duplex:

In this type of communication mode data can be transmitted in both directions, but only in one direction at a time. Both sender and receiver cannot transfer the data at a time. While sending data it cannot receive it and while receiving data it cannot send.

Features:

- Data are transmitted in both direction but single direction at one time.
- Receiving end acts as mere listener while sender sends data and vice versa.
- The communication is slower.
- Willkie Talkie used by the police man is the best example of half-duplex communication mode.

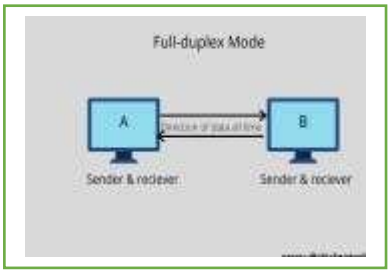


b. Full Duplex

Full Duplex communications allows data to flow the information at the same time. Speaking on telephone of full Duplex communication mode in which both the sender and receiver can speak simultaneously. Bidirectional communication at the same time is called full duplex communication mode.

Features:

- Data are transmitted in both direction at the same time.
- When one end sends data other end can receive as well as send data.
- Communication is faster.
- Telephone is an example of full duplex mode.



S.N.	Simplex	Half Duplex	Full Duplex
1	It is a uni-directional communication	It is a two way directional communication but one at a time.	It is a two way directional communication simultaneously.
2	Sender can send the data but can't receive the data in it.	Sender can send the data and also can receive the data but one at time in it.	Sender can send the data and also can receive the data simultaneously in it.
3	It provides less performance than half duplex and full duplex.	It provides less performance than full duplex.	It providers better performance than simplex and half duplex mode.
4	Examples: radio and newspaper	Examples: Walky-talky and wireless handset	Examples: smart phone and land line phones.

Computer Network:

- It is defined as the collection of two or more autonomous computers which are interconnected together for sharing resources with the help of transmission media and set of protocols.
- A computer network is a logical or physical interconnection between two or more computers such that they could communicate with each other.
- A computer network is the group of two or more computers or computing devices interconnected together for communicating, sharing information, hardware, software and data.

Services provided by the computer network.

- Data sharing
- Print service
- File service
- Database service
- Application service

Advantages of **Computer Network**

1. ***Sharing resources:*** Software and hardware resources such as processor, storage devices, printers, scanner, etc. can be shared among us using computer network. It helps to minimize the operational cost of an organization.
2. ***Saving Cost:*** Sharing of hardware and software resource avoids duplication, helps in optimal utilization of all types of resource like printer, disks, database etc.
3. ***Faster and cheaper communication:*** communication in modern days has become very faster and cheaper to send information to a long distance through network.
4. ***Centralized control:*** all network resources such as computers, printer file, database, etc can be managed and controlled by a central connecting computer also known as the server.
5. ***Enterprises and chain organization developed:*** One office or head of the organization can easily visualize and monitor offices and staffs geographically located in different places through videoconference, CC camera etc.
6. ***Backup and recovery:*** server is used to keep data as backup. It maintains backup of all individual computer information.
7. ***Remote and mobile access:*** a remote user can access resources from the distance using computer network.

Disadvantages of **Computer Network**

1. ***Expensive:*** In order to install computer network, we require some extra cost to purchase networking devices such as hubs, switch, cables, etc.
2. ***Security problems:*** network security is the most challenging job for network administrator in order to protect network resources from authorized users and physical destructions.
3. ***Needs technical person:*** it is very difficult to install and operate good computer network.

Types of Computer Network

On the basis of size computer networks can be classified into three categories:

1. **PAN (Personal Area Network):** The computer network formed around a person is called Personal Area Network (PAN).
   - It is personal device network within a limited area.
   - It consists of a computer, mobile, Personal Digital Assistant (PDA). It can be used for communication among these personal devices for connecting to a digital network and the internet.

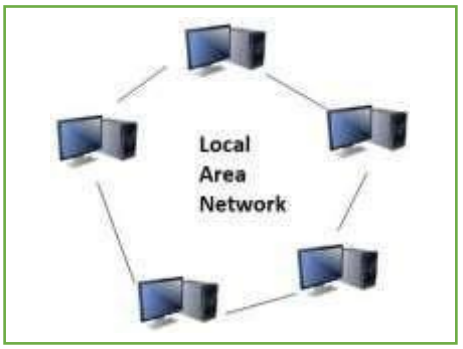
Advantages

1. It can be relatively secure and safe.
2. It offers short- range up to ten meters.

Disadvantages

1. It may establish a bad connection to other network
2. It is limited within shorter distances.

2. ***Local Area Network (LAN):*** A LAN is privately owned small size network. It spans only in small geographical area such as within a room, office, buildings or up to few kilometers (2 to 3 Km). It connects the network resources such as computers, faxes, printers and various networking devices.



Advantages of LAN

1. It is cheaper to establish.
2. Data transmission is faster than MAN and WAN.
3. It has higher security to resources of the network
4. It is eager to establish, manages of the network and operate

Disadvantages of LAN

1. It is limited only to a small area.
2. It can connect less number of computers comparatively.
3. Cannot be used as distributed network.

3. ***Metropolitan Area Network (MAN):*** A MAN can be either public or privately owned network. Its size is bigger than LAN and smaller than WAN. It spans within one metropolitan city or larger geographical area. It can connect large number of computers and heterogeneous multiple LANs within a city maximum, up to 100Km.

Advantages of MAN

1. It covers larger geographical area than LAN.
2. It can connect large number of computer than LAN.
3. We can use guided as well as unguided type of transmission media.

Disadvantages of MAN

1. It is expensive to set up then LAN.
2. Transmission speed slower compared to LAN.
3. It is complex to establish, manage and provides security.

4. ***Wide Area Network (WAN):*** A WAN is basically public type heterogeneous network. It is the largest sized network and connects millions of computers, thousands of LANs, hundreds of MANs around the countries, continents and even the whole world.

Advantage of WAN

1. It covers larger geographical area than LAN and MAN.
2. It can connect large number of computer compared to LAN and MAN.
3. Using WAN communication can be done over a large distance.

Disadvantage of WAN

1. It is expensive to establish, manage and operate.
2. It is the slowest type of network compared to that of LAN and MAN.
3. Highly qualified manpower are required to establish and run these type of network.

Differentiate between LAN and WAN

S.N.	Local Area Network (LAN)	S.N.	Wide Area Network (WAN)
1	Area covered within a local site.	1	Distance up to thousands of K.M.
2	Higher data transfer rates (10 Mbps to 1Gbps even more).	2	Data transfer rate is less
3	It has low error rates.	3	It has higher error rates.
4	It uses simple protocol, low cost devices and low cost installation.	4	It uses complex protocols, expensive devices and high cost installation.
5	It can support limited number of hosts.	5	It can support large number of hosts.
6	Eg. Star, cellular topologies etc.	6	Eg. Internet and intranet
7	Generally, LANs use wireless or digital transmission.	7	WANs use digital or analog signal transmission.

Comparison of LAN, MAN and WAN

S.N.	LAN	MAN	WAN
1	The computer network limited over small geographical area such as building or room.	The computer network spread over the city or country is known as MAN	The computer network spread over the world is known as WAN.
2	It is owned by an private organization	It can be owned by private or public organizations.	There is no single ownership or WAN
3	The bandwidth of LAN is high. i.e. data transmission rate.	The data transmission rate is slower in comparison to LAN.	The data transmission rate is slowest in comparison to LAN and MAN.
4	There is less congestion in LAN.	There is more congestion in MAN.	There is more congestion in comparison to LAN and WAN.
5	It is easy to design and maintain.	It's design and maintenance is difficult than LAN.	It's design and maintenance is difficult than LAN and MAN.
6	There is more fault tolerance.	There is less fault tolerance.	There is also less fault tolerance.
7	It is used in school, college, hospital …etc.	It is used in city, towns.	It is used for countries, continent and globe.
8	Transmission channels are normally twisted pair cables.	Transmission channel are fiber optics, coax cable.	Communication channel ranges from fiber optics cables to communication satellites.
9	It Stands for Local Area Network	It stands of Wide Area Network	It stands for Metropolitan Area Network.

Transmission media:

A transmission media is defined as the means of communication between two networking devices that helps to transfer data from sender to receiver and vice versa.

- Transmission channel is the path through which data are transmitted from source to destination.
- The path through which data transmit form source to destination is known as communication media.

Transmission media is broadly classified into two groups.

1) Bound (guided)/ Wired media

- Guided Media are those communication channel which directly link with each other through cables or other physical media.
- Data are transmitted in closed path through transmission media.
- Data gets transmitted through wires or cables in guided media.
- In guided media there is direct connection between source and destination nodes.
- Guided media has high bandwidth, low cost and high security.
- Data transmission is faster than wireless.
- The most commonly used guided media are: Twisted Pair cable, Coaxial Cable and Fiber Optics.

2) Unbound (unguided) / Wireless media

- The data transmitted in this medium is through electromagnetic waves so that any physical wire or cable is not required for the transmission.
- Unguided media transmission is bounded by geographical areas.
- Different types of unguided media are: Microwave, Radio wave, satellite communication, Bluetooth, Infrared, Wi-Fi, Li-Fi.

S.N.	Guided Transmission Medium	S.N.	Unguided Transmission Medium
1	Signal is directed and contained by the physical limits of the medium.	1	It has no physical medium for the transmission of electromagnetic signals.
2	It is called wired communication or bounded transmission media.	2	It is called wireless communication or unbounded transmission media.
3	The signal energy propagates through wires in guided media.	3	The signal energy propagates through air in unguided media.
4	It types are twisted pair cable, coaxial cable and fiber optic cable.	4	It types are radio wave, microwave and infrared.
5	Examples: Twisted pair cable, coaxial cable and fiber optic cables.	5	Examples: Microwave or radio links and infrared light.

1. ***Wired or Guided Media or Bound Transmission Media:*** The transmission of data and information from source to destination by using physical medium like wires are called bounded transmission media. Its types are as follows.
    1) Twisted pair cable
        i) Shielded Twisted pair cable (STP)
        ii) Unshielded Twisted pair cable (UTP)
    2) Coaxial Cable: 2.1. Thinnet 2.2 Thicnet
    3) Fiber optics

Twisted pair cable: A pair of copper wires is twisted to each other in a helical path making the same structure as a DNA molecule.

- The reason for twisting is to reduce electrical interference.
- It is the cheapest and easily available wire.
- It is mostly used in telephone systems.

Advantages

1. It is cheaper than other cables.
2. It is light and thin. So, it is flexible for LAN.
3. It can travel data in short distance with higher bandwidth.

Disadvantage

1. It is only used for short distance transmission.
2. It can be affected by electrical and magnetic field.
3. It is slower type of transmission media compared to other cables.

STP (Shielded Twisted Pair)

Shielded Twisted Pair cable is identical to UTP but it contains the extra shield covering the each pairs of cables. The installation of STP is comparatively difficult than UTP and they are more expensive than UTP but has high bandwidth.

Features of STP Cable

» Better performance and high data transfer rate.
» Eliminates cross talk
» Faster than UTP
» More expensive

UTP (Unshielded Twisted Pair)

UTP is a commonly used cable because it is easy to install and is suitable for data and voice transmission. Unlike STP, UTP cable doesn't have the extra protection shield.

Features:

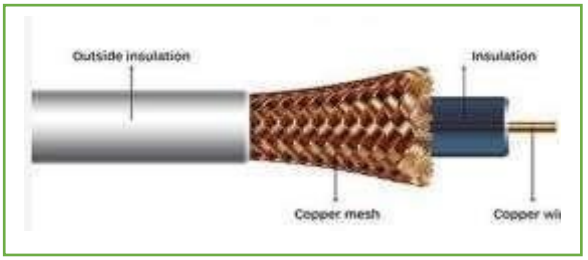
- Less expensive.
- Easy to install.
- High speed capacity.
- Susceptible to external interference.
- Lower capacity and performance than STP.
- Short distance transmission.

S.N.	UTP	S.N.	STP
1	Electromagnetic interference and noise is more in UTP	1	STP cable reduce electrical noise within the cable and from outside of the cable.
2	It offers speed or throughput of about 10 to 1000 Mbps.	2	It offers speed or throughput of about 10 to 100 Mbps.
3	It offers maximum cable length of about 100 meters.	3	It supports maximum segment of length about 100 meters.
4	UTP is widely used for data transmission within short distance and is very popular for home network connecting.	4	STP is mainly used for connection of enterprises over a long distance.
5	The cost of UTP is less when compared to that of STP.	5	STP is costlier than UTP.

Coaxial Cable:

Coaxial cable is a type of copper cable specially built with a metal shield and other components engineered to block signal interference. It is primarily used by cable TV companies to connect their satellite antenna facilities to customer homes and businesses.

Coaxial cable contains two conductors inner and outer, which are separated by an insulator. These two conductors lies parallel to each other. The inner conductor is made up of copper wire which is covered by an inner insulator. The inner insulator is again covered by outer conductor or metal foil. Outer metal wrapping is used as a protection against noise and as the second conductor which completes the circuit. The outer conductor is covered with an insulating cover.



Advantages:

1. It is faster and reliable than twisted pair cable.
2. It can transfer data over medium range of distance.

Disadvantages

1. It is not appropriate for relatively larger distance.
2. It is expensive than twisted pair cable.
3. It is rarely used in computer network.

Coaxial cable is of two types:

(a) **Baseband transmission (Thin net):** It is defined as the process of transmitting a single signal at high speed.

(b) **Broadband transmission (Thick net):** It is defined as the process of transmitting multiple signals simultaneously.

Fiber Optics:

It is the most advanced media in communication, which uses light rather than electricity to transmit information. Optical fiber is very thin media, which is measured in microns and is very hard to identify with our necked eye. They're designed for long-distance, high-performance data networking, and telecommunications. Fiber optic cables support much of the world's internet, cable television, and telephone systems.

Features:

- » Data are transmitted in the form of light signals.
- » It is made up of glasses or plastics cover with fiber to protect.
- » It is difficult to install.
- » It has high bandwidth.
- » Data can transfer longer and faster than twisted pair and coax cable.

Advantages:

1. It has higher bandwidth that means it can handle large volume of data.
2. This medium can be used for long distance transmission.
3. It is the most secured and error free transmission medium.

Disadvantages

1. It is one of the expensive type of transmission media.
2. It is not used for short distance transmission.
3. Highly qualified and technical manpower are required to operate on fiber optics.
4. Difficult to install and Fragile in nature

2. Wireless or Unguided Media or Unbound Transmission Media:

Unbound transmission is also called wireless or unguided media. If there is no physical connectors (wires) between the two communicating device is called wireless transmission media. Its types are as follows.

1. Radio Wave
2. Microwave
3. Infrared
4. Satellite Communication:

1. Radio wave:

Radio wave are the signals ranging frequency in between 3 KHz and 1 GHz.

When the antenna transmits the radio wave; they are propagated in all directions. Due to this the sending and receiving antenna doesn't need to be in line of sight position. Antenna is responsible for converting outgoing data packets into radio waves and vice-versa.

Features:

- It is wireless communication technology that transmits voice or data over the air using a lower frequency band than microwaves.
- The signal is travel in up and down manner so that it can reach any place until the strength of waves falls.
- AM and FM radios and cordless phones use radio waves for transmission.

2. Microwave:

Microwave, in contrast, have been used in data communications for a long time. They have a higher frequency than radio wave and therefore, they can handle larger amounts of data. There are, of course, problems with microwaves attenuation and environmental interference.

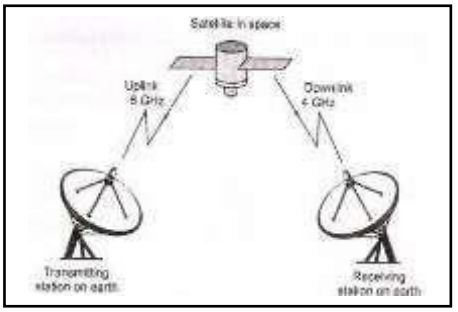
- Microwave transmission is the line of sight transmission.
- The transmit station must be in visible contact with the receive station.
- The microwave is unidirectional.
- This sets the limit on the distance between stations depending on the local geography.
- The microwave is another type of electromagnetic waves that have a frequency range of 1 GHz to 300 GHz.
- In this transmission, whenever the signals are transmitted through an antenna, the signals can make narrow to focus to a particular point. And in this transmission, there is a requirement of proper alignment of sending antenna and receiving antenna.

3. Infrared:

Infrared offers a great unbound photonic solution. Like fiber-optic cabling, infrared communications use light. So, they are not bound by the limitations of electricity. They are normally used for short range communication such as in communication of remote and TV, Ac etc.

Satellite Transmission:

These are the artificial satellites which are placed in the space for the purpose of communication between different antennas available of on the earth. These artificial satellites facilitate the transmission of signals between stations of the earth. They relays and amplifies radio telecommunication by creating communication channel between source transmitter and destination transmitter.



Transmission Impairments Terminologies

(Jitter, Singing, Echo, Crosstalk, Distortion, Noise, Bandwidth and Number of receivers)

Jitter:

- When we make a call over internet with anyone and suddenly line drops and their voice starts breaking up, its called jitter.
- Jitter is defined as a variation in the delay of received packets.
- Jitter is when there is a time delay in the sending of these data packets over your network connection.

Singing:

- Singing is the result of sustained oscillations due to positive feedback in telephone amplifiers or amplifying circuits.
- Singing may be regarded as echo that is completely out of control. This can occur at the frequency at which the circuit is resonant.

Echo:

- Echo in telephone systems is the return of a talker's voice. It is most apparent to the talker himself or herself. Secondarily, it can also be an annoyance to the listener.
- Echo is a major annoyance to the telephone user. It affects the talker more than the listener. Two factors determine the degree of annoyance of echo: its loudness and its length of delay.

Crosstalk:

- Crosstalk is a disturbance caused by the electric or magnetic fields of one telecommunication signal affecting a signal in an adjacent circuit.
- It is an effect a wire on another. One wire acts as a sending antenna and the transmission medium acts as the receiving antenna. Just like in telephone system, it is a common experience to hear conversation of other people in the background. This is known as cross talk.

Distortion:

- Distortion is the change in the form or shape of the signal. Signals are made up of different frequencies are composite signals.
- Distortion occurs in these composite signals. Any change in a signal that alters the basic waveform or the relationship between various frequency components is called distortion. It is usually a degradation of the signal.

Noise:

- Noise is another impairment in data communication. It is another problem of data communication. During data communication there are some random or unwanted signals mix up with the original signal is called noise.
- Noises can corrupt the signals in many ways along with the distortion introduced by the transmission media.

Bandwidth:

- Bandwidth is the amount of data transmitted through the transmission channel at certain period of time. High bandwidth means higher bits of data are transmitted and low bandwidth means less bits of data are transmitted through the communication channel.
- The bandwidth of analog device is measured in Hertz (Hz) and bandwidth of digital device is measured in bps (bits per second). It is also known as data holding capacity of transmission channel.

Receiver:

- Receiver, in electronics, any of various devices that accept signals, such as radio waves, and convert them (frequently with amplification) into a useful form. Examples are telephone receivers, which transform electrical impulses into audio signals, and radio or television receivers, which accept electromagnetic waves and convert them into sound or television pictures.

Network architecture:

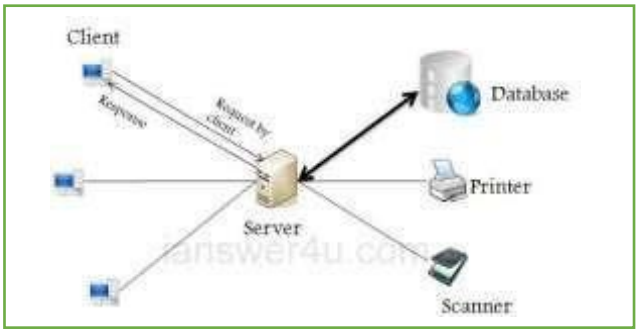
Network architecture refers to the various services provided by the network and it also deals with how data is transmitted from one computer to others.

1. **Client server network**: An arrangement of computers to resource sharing and communicate each other through a central device (server) to all workstations (clients) is called client server architecture one or more computers in the network act as server which provides services to other computers which are called clients.
   - The server is a high capacity high speed computer with a large memory.
   - Server contains the network operating system.

- The central server manages, organize, and coordinate all network clients on the network.
- The most common service is provided by different servers are file services, print services,
- Message services and database services.

Advantages

1. Centralized administration is possible through this network.
2. High security can be provided by suing appropriate server.
3. It is appropriate for large organization.
4. Data recovery and backup process is easier.



Disadvantages

1. If server fails whole network is affected.
2. It is expensive due to use of dedicated server.
3. It is complex to establish and manage.
4. Experienced administrator is required to operate.

2. Peer-peer network: In peer-to-peer architecture computers are connected individually in pair (one-to-one connection). A peer-to-peer network is the type of network in which all computers in the network act both a client and a server i.e. all computers can both request and provide services.

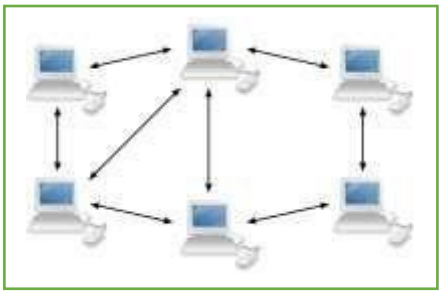
- Each workstations act as both a client and a server.
- There is no central repository for information and no central server to maintain.
- Peer to peer network are generally simpler and less expensive.
- A peer to peer network is also known as a distributed network.
- Peer to peer computing or networking is a distributed application architecture that partitions task or workstations between peers.
- Peers are equally privileged and equal participants in the application.
- Each node shares its sources with other nodes in the network.

Advantages

1. It is simple cheap and easier to set up.
2. Since there is no dedicated server, user can manage their own server.
3. Failure of a computer in a network doesn't effect the other computer in a network.

Disadvantages

1. Data security is very poor in this type of architecture.
2. Data recovery and backup is difficult.
3. It is not appropriate for large scale organization.
4. Network administration is difficult it without dedicated Server.



Difference between Client Server and Peer to Peer

Client Server	Peer to peer
1. It is also known as centralized or server based network.	1. It is also known as distributed network
2. It has central server computer.	2. There is no central server computer.
3. The central server manages, organize, and coordinate all network clients on the network.	3. Peers are equally privileged and equal participants in the application.
4. Client server network are more expensive.	4. Peer to peer network are generally simpler and less expensive.
5. It has high security.	5. It is less security.
6. If server crashes there is a chance of data loss.	6. Data and information is shared around the network, so less chance of data loss.
7. Example, Google server, Yahoo server and Bank etc.	7. Example One to one computer and Bluetooth connectivity etc.

Network or LAN topology and it's types

Network topology refers to the physical layout of the network. It shows the geographical representation of all the links and linking devices, also called nodes. It is the shaped of Network.

The main objectives of the network topology is to find out the most economical and efficient way of transmission channel.

Its types are as:

1. **Bus Topology:** Computers are connected to a single continuous cable that is called 'bus'. It acts as backbone. It is based on client server network architecture.

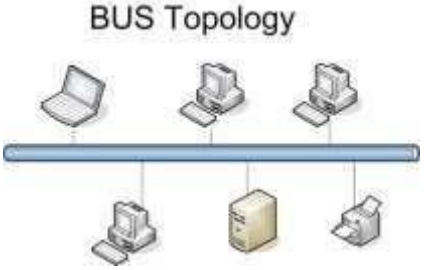


Fig. bus topology

Advantages

- It is simple and easy to setup and extend the network.
- It required less cable.
- If any computer in the network downs, then it does not affect the whole network.
- We can easily connect and disconnect any number of computers in the bus.

Disadvantages.

- Data traffic is very high in bus.
- If there is problem in main cable then entire network goes down.
- It is very difficult to find out the fault in the bus.

2. **Star Topology:-** Computers in the network are connected to each other with the help of central connecting device hub or switch or server. It is based on client server architecture. It is the most popular and widely used topology for LAN.

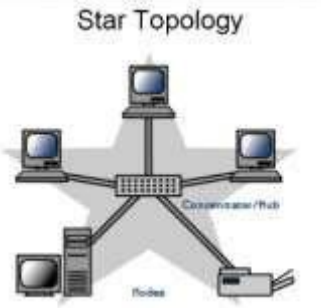


Fig. Star topology

Advantages

- It is simple, reliable and easy to set up and re-configuration.
- It is flexible to connect new computer and remove existing computer in the network.
- It is very easy to find out fault.
- If any computer in the network goes down, then other computers can continue their functions.

Disadvantages

- It requires very large amount of cables.
- It is expensive topology.
- If there is any problem in central device hub or switch then the entire network will be down.
- The data traffic is high in central device hub.

3. **Ring Topology**:- Computers are interconnected to each other by making a closed circular structure that means each computer is connected to other two adjacent computer in either network architecture.

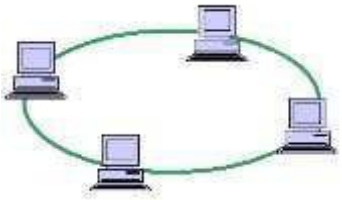


***Fig**. Ring Topology*

Advantages

- It simple and inexpensive topology.
- There is less chance of data collision because of unidirectional data transmission.
- There is no server so each computer has equal access facilities to the resources.
- Its performance is better than bus topology for small size network.

Disadvantages

- It is not flexible topology so it is difficult for adding and removing new nodes.
- It is not suitable for large size network.
- If there is problem in any computer or connection then the entire network goes down.
- It is very difficult to find out the errors in the network.

4. **Mesh Topology:-** Every computer in the network has point to point connection to all other computers by using multiport connector. It is also based on peer to peer architecture.

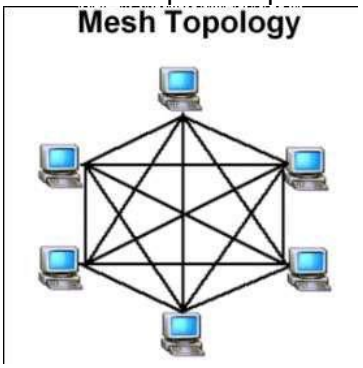


Fig. mesh topology

Advantages

- It is fastest and most reliable topology.
- Failure in any computer or transmission media does not affect the rest of the network.
- There is less amount of data traffic due to multiple paths.

Disadvantages

- It is very much complex and most expensive topology.
- It is difficult to find an error in the network.
- It is difficult to add and remove nodes in the network so it is not flexible.
- It requires maximum amount of cables and multiport connectors.

5. **Tree Topology:-** Tree topology is the extension of bus or star topology. Data can flow from top to bottom and vice versa.

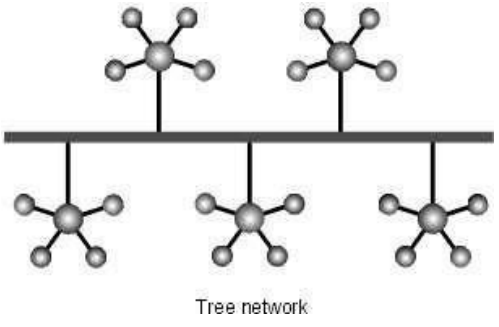


Fig. Tree Topology

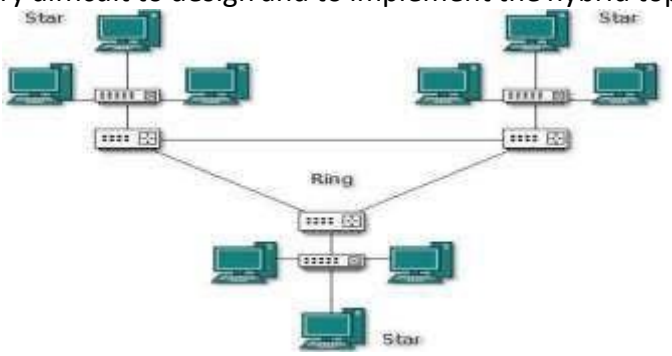
Advantages

- It is easy to manage network as per our needs.
- It is very flexible so we can add and remove any number of nodes.
- It is easier to find the fault nodes or hubs in the network.

Disadvantages

- The failure of root node will cause the failure of entire network.
- It is expensive because it needs large number of cable and network device.
- The data traffic is high at root nodes.

6. ***Hybrid topology:*** if two or more topologies are combined together then it is called hybrid topology. So it is very difficult to design and to implement the hybrid topology. It is expensive too.



Advantages

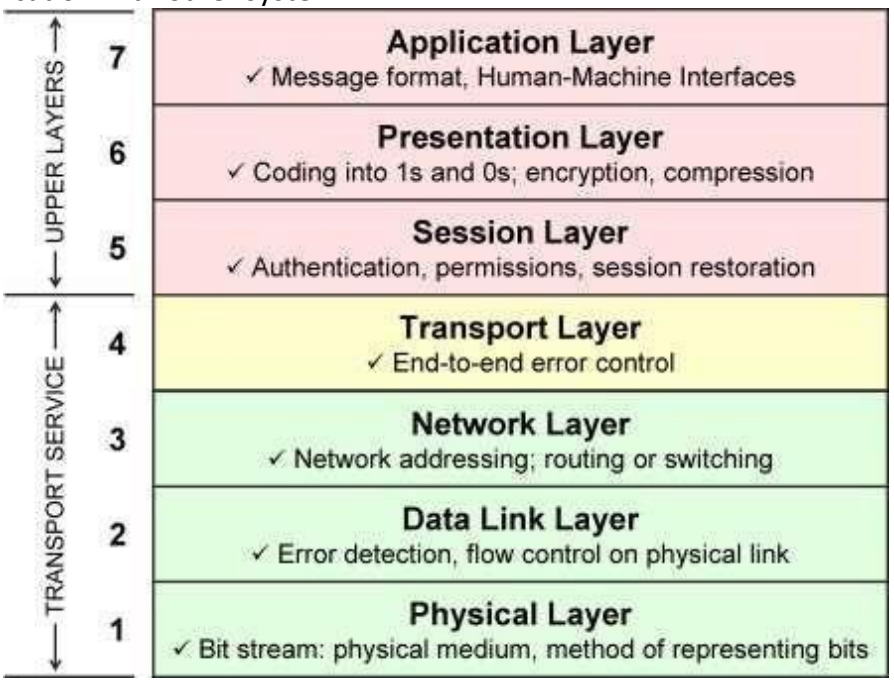
- Reliable
- Scalable
- Flexible
- Effective

Disadvantages

- Complexity of design
- Costly connecting devices
- Costly infrastructure

OSI (Open System Interconnection) reference model

It is based on a proposal developed by the international organization for standardization (ISO). The model is called ISO OSI reference model, because it deals with connecting open system i.e. the system that are open for communication with other system.



1. **Physical Layer**: This layer concerned with transmission of bit it determines voltage level for 0 & 1. It also determines the data rate of the system. This layer involves standardized protocol dealing with electrical & signaling interface.
2. **Data Link Layer**: It handles error in physical layer. This layer ensures the correct delivery of frame to the destination address. It consists of 2 parts or 2 sub-layers. i.e.
   i. Logic Link Control
   ii. Media Access Control
3. **Network Layer:** This layer is concerned with transmission of packet. N/w layer protocol chooses the best path to send a package called routing. Two protocols are widely used in n/w layer.
   i. X.25 Protocol
   ii. Internet Protocol
4. **Transport Layer:** It provides the mechanism for the exchange of data between end systems. It ensures that the data received is in fact in order. Following jobs are performed by this layer.
   i. Port Addressing
   ii. Segmentation & Reassemble
   iii. Connection Control
5. **Session Layer:** It is responsible for requesting logical connection to be established for communication process. This logical connection is termed as session. It also provides data synchronization between two communication terminals.
6. **Presentation layer:** This layer translates format data to adapt to the needs of the application layer & nodes at both receiving & sending end of communication process. It handles data communication, formatting, encryption, decryption, etc.
7. **Application Layer:** It is the top-most layer of OSI model & provides user access to the n/w. It provides services that support user application, such as database access, email & file transfer, etc.

Network Connecting devices:

The devices of computer network which are used to connect network is called the network connecting devices. We can connect two or more networks together to create larger network. A LANs can be connected others LANs, MANs or WAN.

4. Network Interface Boards or Network Interface Card (NIC):

The NIC contains the electronic circuitry needed to ensure the reliable communications between workstations and servers. The NIC is the electronic interface between the computer and the LAN cabling. The card itself uses a bus-specific edge connector to plug into the computer motherboard. On the exposed side of the NIC, there are cabling ports that plug directly into the LAN cabling.

5. Repeaters:

As the name implies, repeaters repeat network data. In general, repeaters operate at the electronic level and contain no real intelligence. A repeater accepts weak signals, electrically regenerates them and then sends the messages on their way. There are two types of repeaters: amplifiers and signal-regenerating repeaters.

3. Hubs:

Technically speaking, a hub is simply a multi-port repeater. In addition to regenerating network data, hubs add form and function to the layout of the LAN. In many topologies, the hub is the central component of the network transmission media. There are three types of hubs. They are the passive hub, active hub and intelligent hub.

4. Bride:

It is a device, which connects different network segments and passes data with the same communication protocols. It is the connecting device between two or more hubs.

- It operates at layer 2 (Data Link Layer) of OSI Model.
- It reduces unnecessary traffic problem by controlling broadcasting.
- Bridges are more intelligent than hubs because they maintain MAC address tables in them and forward data looking on it.

- Bridge collect and pass packets between 2 networks segments.

5. Router:

A router transmits information from one network to another. The router selects the best path to route a message, based on the destination address and origin. The router can direct traffic to prevent head-on collisions and smart enough to know when to direct traffic along back roads and shortcuts.

- Direct signal traffic efficiently.
- Route message between any two protocols.
- Route message between different topologies.
- Route message across fiber optic, coaxial, and twisted-pair cabling.

6. Gateways:

Gateways are just like routers but much more complex and powerful than routers. They are slower than router and expensive. A gateway has all the features of router and bridges but it can translate instruction set on sending network into receiving network.

- Gateways make communication possible between different architecture and environments.
- Gateways having higher layer functionalities and works on multiple layer protocols.

7. Wi-Fi:-

The term Wi-Fi suggests wireless Fidelity. It is hardware and software devices. It is wireless technology and network connecting device. Wi-Fi is not a technical term. The technical term of Wi-Fi is "IEEE802.11"

- It has limited range.
- Wi-Fi is used in many personal computers, video game consoles, MP3 players, smart phones, printers, digital camera, laptops computers and other devices.
- Wi-Fi is used to create wireless LAN to connect computer system.

8. Bluetooth:

Bluetooth is a wireless technology standard for exchange data over short distance from fixed and mobile devices. Bluetooth is used to create personal area networks (PANs) with high levels of security.

9. Infared (IR):

Infared (IR) light is electromagnetic radiation with a wavelength longer than that of visible light and below the red light. These wavelengths correspond to a frequency range of approximately 100 GHz to 100 THz and include most of the thermal radiation emitted by objects near room.

Some basic terms

Internet

The Internet is an interconnected network of thousands of networks and millions of computers linking business, education institutions, government agencies, and individuals together.

Used / application of internet

1. Search information
2. E-mail service
3. Communication
4. File Transfer
5. Remote login
6. Publishing of articles, reports, and newsletter.
7. Online education
8. Online shopping.
9. Entertainment.

Sub Net Mask

The subnet mask number helps to define the relationship between the host (computers, routers switches, etc.) and the rest of the network. As the name indicates, the subnet mask is used to subdivide a network into smaller, more manageable chunks.

Gateway

A gateway is a network node that servers as an access point to another network, often involving not only a change of addressing but also a different networking technology.

IP Address

An Internet Protocol address (IP Address) is an unique identification number assigned to each devices connected in a network that uses Internet protocol for communication. The typical IP address looks like 216.27.67.137

IP address are expressed in decimal as above which is easier for us to remember, but computers communicate in binary. So same IP address looks like

11011000.00011011.00111101.10001001

IP address has four sets of number separated by a period (.), These sets are Called Octets. IP address perform mainly two functions. One is identification of Network and Identification of particular host in that network.

MAC address

Media Access Control (MAC) address also known as hardware or physical address is unique number associated with a network adapter (NIC). It is used to uniquely identify each device (node) of a network. MAC address is usually assigned by the manufacturer of a network interface card (NIC) and are stored in its ROM. All NIC developed contains unique MAC address. It is a 12 digit hexadecimal number (48 bit in length) and are Witten in following formats.

MM:MM:MM:SS:SS:SS or MM-MM-MM-SS-SS-SS

The first half of the address contains ID number of the adapter regulated by an Internet standards body and other half represent the number assigned by the manufacturer. For example: 00:A0:C9:14:C8:29 The prefix 00A0C9 indicates manufactures Intel Corporation and 14C829 indicates a particular host.

Intranet:

An intranet is a private network that uses internet protocols, to securely share part of an organization's information between its employees. An intranet is a Local Area Network or Wide Area Network that uses TCP/IP protocol but, belongs to a corporation, school, or organization.

The intranet is accessible only to the organization's workers. If the intranet is connected the Internet, then it is secured by a firewall to prevent unauthorized users from gaining access to it.

Advantages of an Intranet

1. **Workforce productivity:** Employees can easily access and share information in-between their workgroups which enhances productivity

2. **Time:** With intranets, organizations can make more information available to employees in any time.

3. **Communication:** Intranets can serve as powerful tools for communication within an organization.

Extranet:

An extranet is a private intranet that can be accessed by outside users over the secure channel. To gain entrance to the extranet's resource, an external user must log on to the networks by providing a valid user ID and password. The extranet is a combination of the public Internet and the closed intranet.

Difference between Internet and Intranet?

Intranet	**Internet**
It is a private network.	It is a public network.
Intranet users are your own employees who know a lot about the company, its organizational structure and special terminology.	Internet user know much less about your company and also care less about it.
The intranet is used for everyday work inside the company.	The Internet is mainly used to find out information.
The intranet will have many official draft reports, project progress reports, human resource information, and other detailed information.	Internet have all types of information based on requirements.
Intranet has less amount of information.	Internet has tremendous amount of information.
Intranet can work on low and mid-bandwidth.	Internet requires higher band width.

Network Tool

Network tools are used for supporting easier and effective network connections. Some of the network tools are:

1. **Packet tracer:**
   Packet Tracer is an innovative and powerful network simulator that can be used for a practice build own network with routers, switches, wireless, and much more. It allows to experiment with network behaviour, build models and to ask "what if" questions. It is used to trace the movement of data packets in data communication. Packet Tracer provides simulation, visualization, authoring, assessment, and collaboration capabilities to facilitate the teaching and learning of complex technology concepts.

2. **Remote login:** Remote login 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. Remote Login is a process in which user can login into a remote site i.e. computer, and use services that are available on the remote computer. With the help of remote login a user can understand result of transferring result of processing from the remote computer to the local computer.

Protocols:

A communication protocol is a formal description of digital message formats and the rules for exchanging that message in between computer systems. Protocols define a set of formal rules describing how to transmit data especially across a network.

1. TCP/IP (Transmission Control Protocol/Internet Protocol): TCP/IP is a layered set of protocols TCP is reliable, but complex transport-layer protocol. It is stream connection-oriented and reliable transport protocol. TCP/IP is the protocol used by Internet. It adds connection-oriented and reliability features. TCP is responsible for making sure that the data is transmitted to other end. It keeps track of what is sent, and re transmits any data that has not reached its destination. The Internet Protocol (IP) is the principal communication protocol used for transmitting data packets across and network using the Internet Protocol Suite. Internet exist due to TCP/IP.

2. FTP (File Transfer Protocol): This protocol is used for transferring data between client and server over TCP/IP (Internet). Hence, it is responsible for uploading and downloading files to and from the server.

3. UDP (User Datagram Protocol): UDP is simple, connection less, unreliable transport protocol. It performs very limited error checking. It is mainly used for transmitting multimedia data, which requires faster transmission and error checking is not used.

4. SMTP (Simple Mail Transfer Protocol): SMTP is a standard protocol for transmitting electronic mail (email) by the internet. It is a Internet mail protocol. It is a TCP/IP protocol used to send emails

5. POP (Post Office Protocol): POP is also a protocol for transmitting email. It is simple but has limited functionality. It is an application layer Internet standard protocol used by clients to access e-mail from a server over a TCP/IP connection i.e. internet. POP3 (POP version 3) is used at present. POP3 is supported by most modern webmail services such as Gmail and Yahoomail.

6. HTTP (Hypertext Transfer Protocol): HTTP networking protocol for distributed, collaborative, hypermedia information systems. It is used transmitting hypertext or HTML based document. It is the foundation of data communication for the World Wide Web used by web browser to communication with respective servers.

7. HTTPS (Hypertext Transfer Protocol Secure): HTTPS is combination of Protocol with the SSL/TLS protocol to provide encrypted identification of a network. HTTPS connections are often used in World Wide Web for sensitive transactions. The main objective of HTTPS is to create a secure channel over an insecure network

Layer	**Protocols**
Application	FTP, TFTP, Telnet, Gopher, HTTP
Presentation	SMTP, NFS
Session	Telnet, SNMP
Trasport	TCP, UDP
Network	IP
Data Link	ARP, RARP, RIP
Physical	Ethernet, Token Ring, FDDI

Bibliography

- Balaguruswamy, E. Programming in ANSI C. Tata McGraw-Hill
- Evans, P. (2001) Information and Communication Technology, 2 edition Payne Call way Publishers.
- French, C.S. (1996). Computer Science, 5% edition, W.M. Print Ltd.
- Gottfried, B. Programming with C, Tata McGraw-Hill,
- Jaiswal, S. (2001). Information Technology Today, 3rd edition, Galgotia Publication
- Kanetkar, Y. Let Us C, 8th edition, BPB Publication
- Lafore, R. Object Oriented Programming in C++, Golgotia Publication.
- Lafore, R. Turbo C. Golgotia Publication.
- Laudon, K.C. & Laudon, J.P. Management Information Systems, Managing Digital Firm, 9 edition, Prentice-Hall, India. Shrestha, P.R., Bhusal, R., Khanal, R. (2077). Essentials of Computer Science. Kathmandu Asmita Books Publisher and Distributors (P) Ltd.
- Silberscratz, HF & Korth, S. Database System Concepts, 4th edition, Tata McGraw-Hill
- Stephen, D., (1999). Information Systems for You, 2nd edition, Stanley Thornes (Publishers) Ltd.
- Tanenbaum, A.S., Computer Networks, 4th edition, Prentice Hall,
- William, S. Data and Computer Communication, 6th edition, Prentice Hall

Web References:

- https://www.javatpoint.com
- https://www.w3schools.com
- https://www.tutorialspoint.com
- https://www.google.com
- https://www.wikipedia.org

The End

COMPUTER SCIENCE

Grade: XII

WEB TECHNOLOGY II

REFERENCE NOTE

NEB Important Questions for Computer Science XII

Unit 3- Web Technology II

1. What is web technology? Explain different data types used in JavaScript.
2. ***Differentiate between client side scripting and Server-side scripting.***
3. What is event handling in JavaScript? Give one example.
4. ***What is JQuery? Write its features and Write a program to displaying a message "Hello Class 12" using JQuery.***
5. What is JavaScript? How can you add JavaScript to an HTML page? Describe with example.
6. What is jQuery? Write its features.
7. ***What is Java Script function? Explain with calling function examples.***
8. What is PHP? Write the advantages of PHP.
9. Explain the different operators used in PHP.
10. ***What are the features of PHP? Write a PHP program to display the largest among three numbers.***
11. Define SQL? Write down the SQL queries to create a database, create a table in the database, insert data in the table, and query the data to display it.
12. Write down the server side script to create a database, connect with it, create a table and insert data in it.

Practical Programs

1. Write a java Script to display 1 to 10 using for loop, while loop and do while loop.
2. Create a Page with a button with value "Computer" on clicking the button your page should" Computer Science".
3. ***Write a JavaScript program to display "Welcome Class-12" using onload event.***
4. Design a form with username, address, e-mail, password and submit button. Validated the form using jQuery.
5. ***Design a form with username and password and submit button. Write a PHP code to get value of username and password using a) $_POST variable and b) $_GET variable.***
6. Write a PHP program to check if a string is a palindrome or not.
7. ***Write a Java Script program to calculate the factorial of a given number.***
8. Write a java script program to input three number and find largest one using java and PHP.

Web technology

Introduction

Web Technology is the tools and techniques which enables two or more computing devices to communicate over a network i.e. Internet.

Web Technology consist of two words, the web refers to the World Wide Web generally known as World Wide Web.

WWW is the cyber space containing webpages, documents, and any other resources which are identified and located with the help of their URLs.

Technology refers to the tools and techniques that makes these resources available on the Web such as, web browsers to view content of web, Programming languages and frameworks for the development of websites, Database to store data at back end, protocols for communicating on the web, multimedia elements etc.

Web development

It is the process of designing and developing website which are hosted through internet or intranet. The process of developing web can range from developing static page to a complex such as web based application social media sites, E-commerce.

Web development includes web design, web content development, client side scripting, server side scripting, web engineering etc. Since, web development consists of several inter-related task which can be accomplish by different types of developer who focuses on different aspect of web creation.

Frontend	**Backend**
Client Side	Server Side
Website Design	Database
HTML	PHP
CSS	Java
JavaScript	Python
AJAX	Ruby
UI/UX	Servers
Some UI technologies	Some Backend technologies
Ruby	.NET

Scripting Language:

- JavaScript is a scripting language. A scripting language is a lightweight programming language. 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. JavaScript and Java are completely different language, both in concept and design.

Unlike HTML, JavaScript is case sensitive-therefore watch your capitalization closely when you write JavaScript statements, create or call variables, objects and functions.

A scripting language is a programming language designed for integrating and communicating with other programming language. It is used in server side as well as client side. Some of the most widely used scripting language are JavaScript, VBScript, PHP, Perl, Python, Ruby etc.

Application Areas of Scripting Language

- Dynamic web applications
- Game application and Multimedia
- Scripting like Ruby 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 Language 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: P**ractical **E**xtraction and **R**eporting **L**anguage, 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 is 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.

S.N.	Scripting Language	S.N.	Programming Language
1	Scripting language are platform-specific.	1	Programming language are platform-independent.
2	Most of the scripting languages are interpreted.	2	Most of the programming language are complied.
3	Scripting language runs slower than programming language.	3	Programming languages runs faster than scripting language.
4	Developer has to write less code compared to programming language.	4	Developer has to write much code compared to scripting language.
5	We cannot create standalone application with scripting language only.	5	We can create standalone application with programming language only.
6	Examples, JavaScript, VBScript, Python, Perl, ASP etc.	6	Examples, C, C++, Java etc.
7	Scripting languages run inside other programs. It is dependent on other programming language.	7	It is not dependent on other programs to run. It is independent.

Server side and client side programming

Client-Side Scripting programming

Client-side scripting is performed to generate a code that can run on the client side i.e (front end) browser without needing the server-side (back end) processing. Basically, client-side scripts are placed inside an HTML document.

The client-side scripting can be used to layout the content of the web. For example, when a user makes a request through web browser for a webpage to the server, it just sent the HTML and CSS as plain text, and the browser interprets and renders the content of web in the client end (user).

Client-side scripting is designed to run as a scripting language which can be executed by web browser. Front end developer is someone who design and develop client side of a website. Generally he or she works in user interface (UI) of a website. Front end developer must be at least fluent in three different languages i.e. HTML, CSS, JavaScript whereas, there are several other libraries which can be used for front end development.

Advantages of Client side Scripting:

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

Disadvantages of Client side Scripting:

1. All browsers may not support client side script
2. The code is not secure because anyone can look at the code
3. Users can disable the client side scripts so required content may not be displayed
4. Database connection is not possible with client side scripting.
5. Dynamic content cannot be displayed.

Server-Side Scripting programming

Server-side scripting also known as back-end runs on the server where the application is hosted. Server-side is used to serve content depending upon the user request. Back end helps to create dynamic web based application that allows user to interact and communicate with the application. Back end language also helps to connect front end with data base. So that, User can store and retrieve data as per the requirement. Back-end developer is responsible for server-side programming. Some of the popular server-side (back-end) scripting language are ASP, JavaScript (using SSJS (Server-side JavaScript e.g. node.js), Perl, PHP, Ruby, Python etc.

The client-side scripting emphasizes making the interface of the web application or website (UI) more appealing and functional. Whereas, server-side scripting emphasizes on data accessing methods, error handling and fast processing etc.

Advantages

1) You can create dynamic pages.
2) Can connect to database that resides on the web server.
3) Can access files from the server to client browser Users are not able to block the contents from server
4) The actual code is not visible to the client
5) Authentication and verification of user is possible
6) It supports many databases like MySQL Oracle,
7) Efficient storage and delivery of information
8) Customized user experience.
9) Controlled access to content
10) Notification and communication
11) Users do not need to download plug-in like java or flash
12) The content management system (CMS) makes editing simpler

Disadvantages of Server side Scripting

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

Note: *Full-stack developer understand both Front end and back end development process. They can accomplish entire project. Full stack developer must have expertise in client site and server site Scripting language. Moreover, he/she has a great knowledge of integrating database with the application.*

```
<html>
<head>
<title> Inline javascript Example</title>
<body>
        <form>
<center><input type="button" value="ClickMe" onclick="alert('Button Clicked: ')">
        </center></form>
</body>
</html>
```

Comparison between Client-side and Server-side Scripting

This section elaborates the fundamental differences between client-side and server-side scripts:

1. The client-side script is executed at the front-end in the client's browser while the server-side script is executed at the back end with a web server.
2. The client-side script is visible to the user of the web browser while the server-side script is hidden.
3. The client-side script is not secure while the server-side script is secure.
4. The client-side script does not need to interact with the server while the server-side script needs a web server to be processed.
5. The client-side script is executed on a local computer while the server-side script is executed on a remote computer.
6. The client-side script has a faster response time than the server-side script.
7. Client-side script is executed after the browser receives the web pages sent by the server while the server-side script cannot execute the client-side script.
8. The client-side script cannot connect with the database while the server-side script can connect with the database present on the server-side.
9. The client-side script cannot access the files while the server-side script can access and manipulate the files present at the webserver.
10. The client-side script helps create interactive web pages while the server-side script helps create web pages with dynamic data.

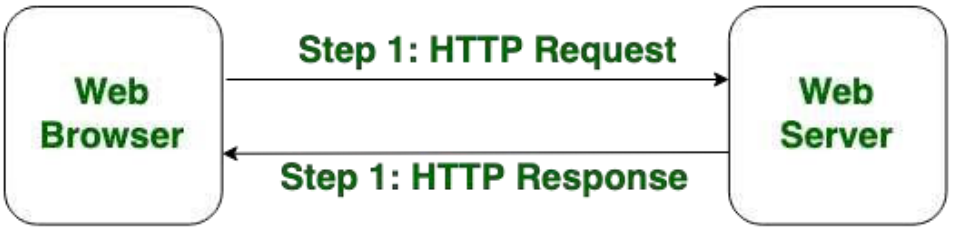
S.N.	Server Side Scripting	S.N.	Client side scripting
1	The Server executes the server side scripting.	1	The client (web browser) executes the client side scripting.
2	It can be used to connect database on the web server.	2	It cannot be used to connect to the database on the web server.
3	Server side scripting response is slower.	3	Client side scripting response is faster.
4	Source code is not visible to the user so it is secure.	4	Source code is visible to user so it is relatively insecure.
5	Users can not block server side scripting.	5	Users can block client side scripting
6	Examples: PHP, ASP.NET, ASP, Ruby on rails, Python, Perl.	6	Examples: JavaScript, VBScript
7	It does not depend on client. Any server side technology can be used.	7	It depends on browser and version of the browser.

Internet Technology

The Internet is the global system of interconnected computer networks that uses the Internet protocol suite 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, electronic mail, telephony, and file sharing.

Static Website:

Static Web pages are very simple. It is written in languages such as HTML, JavaScript, CSS, etc. **Static websites** are the websites that doesn't change the content or layout dynamically with every request to the web server. Static websites display exactly the same information whenever anyone visits it. User sees the updated content of Static Website only when a web author manually updates them with a **text editor** or any web editing tool used for creating websites. Static webpages do not have to be simple plain text. They can feature multiple design and even videos.



Static Web Page

Features of Static Websites:

- Static websites are the simplest kind of website that you can build.
- Every viewer will see the exactly same text, multimedia design or video every time he/she visits the website until you alter that page's source code.
- Static websites are written with the help of **HTML and CSS.**
- The only form of interactively on a static website is **hyperlink**.
- Static website can be used for the information that doesn't change substantially over months or even years.
- Static pages are easy and simple to understand, secure, less prone to technology errors and breakdown and easily visible by search engines.
- **HTML** was the first tool with which people had begun to create static web pages.
- Static websites provide **flexibility**.
- **Lightweight**.
- Static websites perform **faster** and **well** than dynamic ones.

Advantages of Static websites

- Static websites are highly cost-effective for publishing.
- They require less coding and technical knowledge.
- Static websites are easier to make.
- Static websites are quick to develop.
- Static websites are cheap to host.
- A static website contains data which is immutable.
- Static websites are beginner level. A programmer with knowledge of HTML, CSS, and JavaScript can build static websites.
- It's easy to create and host online.
- Static websites provide security.

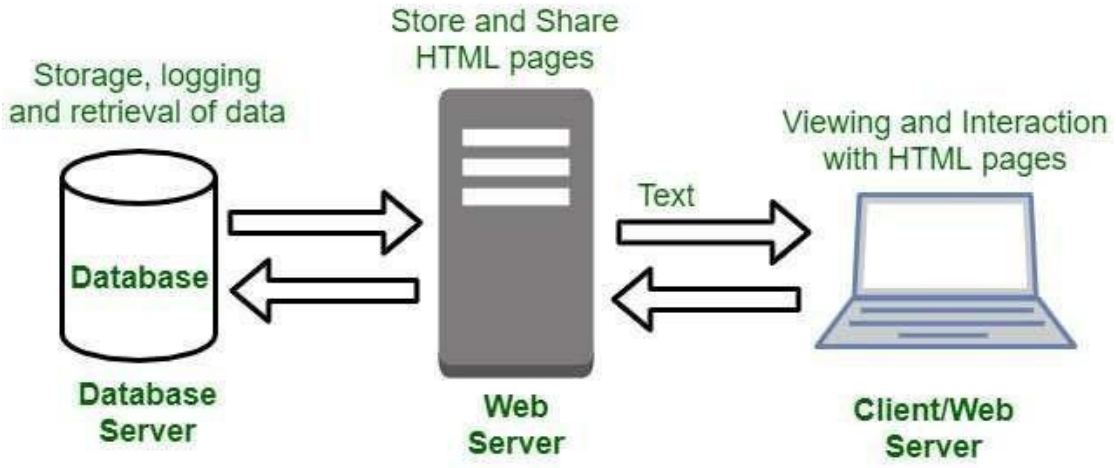
Disadvantages of Static Websites:

- Requires web development expertise to update site.
- Site not as useful for the user.
- Content can get stagnant.
- Send the same response for every request.
- Dynamic functionality works only on the client side.

Dynamic website:

Dynamic websites are those websites that changes the content or layout with every request to the webserver. These websites have the capability of producing different content for different visitors from the same source code file. There are two kinds of dynamic web pages i.e. client side scripting and server side scripting. The client-side web pages changes according to your activity on the web page. On the server-side, web pages are changed whenever a web page is loaded.

Example: login & signup pages, application & submission forms, inquiry and shopping cart pages. A Typical Architecture of dynamic website



There are different languages used to create dynamic web pages like PHP, ASP, .NET and JSP. Whenever a dynamic page loads in browser, it requests the database to give information depending upon user's input. On receiving information from the database, the resulting web page is applied to the user after applying the styling codes.

Features of dynamic webpage:

- These websites are very flexible.
- In these websites the content can be quickly changed on the user's computer without new page request to the web browser.
- In these websites the owner have the ability to simply update and add new content to the site.
- These websites are featured with content management system, e-commerce system and intranet or extranet facilities.
- Most of the dynamic web content, is assembled on the web using server-scripting languages.

Advantages of dynamic webpage:

- It provides more functional websites.
- It is very easy to update.
- It helps in the search engines because new content brings people back to the site.
- These are interactive websites because these can be customized.
- These websites can work as a system to allow staff or users to collaborate.

- It's easy to modify or update data.
- It provides a user-friendly interactive interface for users.
- Proves smooth navigation.
- provide interactive user interface
- It provides a better user experience.
- It provides real-time data.

Disadvantages of dynamic webpages:

- These types of websites are complex.
- These are more expensive to develop.
- Hosting of these websites is also costlier.
- It requires a rapid, high-end web server.
- High production costs.
- Slow to load content.
- Client will require a skilled programmer to build a dynamic website.
- Hosting a website is costly as compared to a dynamic website.
- Low speed compared to a static website

Application of Dynamic Website:

- Online booking system:
- E-commerce website.
- Voting or polls,
- Forums
- E-newsletter.

JavaScript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

This tutorial has been prepared for JavaScript beginners to help them understand the basic functionality of JavaScript to build dynamic web pages and web applications.

Features

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- It is case sensitive language.
- JavaScript is supportable in operating system.
- It provides good control to the users over the web browsers.

```
<script>
document.write("Hello JavaScript by JavaScript");
</script>
```

Advantages of JavaScript

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 users' 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.

Uses of JavaScript

- Client side validation
- Dynamic drop down menus.
- Displaying date and time.
- Displaying pop-up windows and dialog boxes
- Displaying clocks.
- Event handling.
- Developing Mobile applications
- Creating web browser based games.
- Building web servers
- Adding interactivity to website.

Adding JavaScript to HTML

JavaScript, also known as JS, is one of the scripting (client-side scripting) languages, that is usually used in web development to create modern and interactive web-pages. The term "script" is used to refer to the languages that are not standalone in nature and here it refers to JavaScript which run on the client machine.

In other words, we can say that the term scripting is used for languages that require the support of another language to get executed. For example, JavaScript programs cannot get executed without the help of HTML or without integrated into HTML code.

1. Embedding code
2. Inline code
3. External file

1. Embedding code:-

To add the JavaScript code into the HTML pages, we can use the <script>. .... </script> tag of the HTML that wrap around JavaScript code inside the HTML program. Users can also define JavaScript code in the <body> tag

(or we can say body section) or <head> tag

because it completely depends on the structure of the web page that the users use.

```
<html>
<head>
<title> page title</title>
<script>
document.write("Welcome to Java Script Programming");
</script>
</head>
<body>
<p>In this example we saw how to add JavaScript in the head section </p>
</body>
</html>
```

We can also define the JavaScript code in the <body> tags or body section.
Let's understand through an example.

```
<html>
<head>
<title> Body Section Example</title> </head>
<body>
<script>
document.write("Welcome to Javatpoint");
</script>
<p> In this example we saw how to add JavaScript in the body section </p>
</body>
</html>
```

2. Inline code:-

Generally, this method is used when we have to call a function in the HTML event attributes. There are many cases (or events) in which we have to add JavaScript code directly eg., OnMover event, OnClick, etc. Let's see with the help of an example, how we can add JavaScript directly in the html without using the **<script>. </script>** tag.

Let's look at the example.

```
<html>
<head>
<title> Inline code example</title>  </head>
<body>
<p>
<a href="#" onClick="alert('Welcome !');">Click Me</a>
<button onclick="alert('Hello Class 12')">Click Here</button>
</p>
<p> In this example we saw how to use inline JavaScript or directly in an HTML tag.
  </p>   </body>  </html>
```

3. External file:-

We can also create a separate file to hold the code of JavaScript with the (.js) extension and later incorporate/include it into our HTML document using the **src** attribute of the <script> tag. It becomes very helpful if we want to use the same code in multiple HTML documents. It also saves us from the task of writing the same code over and over again and makes it easier to maintain web pages.

In this example, we will see how we can include an external JavaScript file in an HTML document.
Let's understand through a simple example.

```
<html>
<head>
<title>Including a External JavaScript File</title>
</head>
<body>
<form>
<input type="button" value="Result" onclick="display()"/>
</form>
<script src="hello.js">
</script>
</body>
</html>
```

Now let's create separate JavaScript file = Hello.js

```
function display()
 {
alert("Hello Friends!");
}
```

JavaScript Fundamentals

Script

JavaScript statements are written within <script>…..</script>. The <script> tag alerts the browser program to start interpreting all the statements between these tags as a script. A simple syntax of your JavaScript will appear as follows:

Syntax: <script>
block of Statements
</script>

Statements:

JavaScript statements are composed of Values, Operators, Expressions, Keywords, and Comments. The program consists of many statements. The statements are executed, one by one, in the same order as they are written. It is often called JavaScript code.

var area = l*b;

JavaScript comments

The **JavaScript comments** are meaningful way to deliver message. It is used to add information about the code, warnings or suggestions so that end user can easily interpret the code.

The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

There are two types of comments in JavaScript.

1. Single-line Comment
2. Multi-line Comment

1. Single line Comment

It is represented by double forward slashes (//). It can be used before and after the statement.

Let's see the example of single-line comment i.e. added before the statement.

```
<script>
// It is single line comment
document.write("hello javascript");
</script>
```

Let's see the example of single-line comment i.e. added after the statement.

```
<script>
var a=10;
var b=20;
var c=a+b; //It adds values of a and b variable
document.write(c); //prints sum of 10 and 20
</script>
```

2. Multi line Comment

It can be used to add single as well as multi line comments. So, it is more convenient.

It is represented by forward slash with asterisk then asterisk with forward slash. For example:

```
/* your comment here  */
```

It can be used before, after and middle of the statement.

```
<script>
/* It is multi line comment.
It will not be displayed */
document.write("Example of javascript multiline comment");
</script>
```

Whitespace and Line Breaks:

JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs. To make a code readable and understandable, formatting and indenting the code by using several spaces, tabs, and newlines can be used freely in the program.

Case Sensitivity:

JavaScript is a case-sensitive language. The keywords, variables, functions names, and any other identifiers must always be typed with a consistent capitalization of letters. So the identifiers Time and TIME will convey different meanings in JavaScript.

Output:

Java Script can "display" data in different ways: It defines the ways to display the output of a given code. The output can be display by using four different ways which are listed below:

Basic Display function in JavaScript

1. Document.wirte

Using document.write( ), the output can be displayed in browser page.

```
<html>
<head>    <title> Display functions in Javascript </title>
</head>
<body>
    <h1> Output with document.write</h1>        <br>
    Hello Class 12 <br>
    <script>
        document.write(7+5)
    </script>
    <input type="button" value="Click Me" onclick="document.write(7+5)">
</body>
</html>
```

Inner HTML

Using inner HTML, the output can be displayed into the HTML element. The innerHTML property sets or returns the HTML content (inner HTML) of an element.

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

2. Window. Alert

Using window.alert( ), the output can be displayed into an alert box. window.confirm() and window.prompt() are also other method to output into an alert box.

```
<html>
<head>        <title> Display functions in Javascript </title>
</head>
<body>
        <h1> Output Message with window alert</h1> <br>
        <script>
            alert(7+5);
        </script>
</body></html>
```

3. Console.log

Using console.log( ), the output can be displayed into the browser console. To use console of web browser, right click in web browser and select inspect. The console will display.

```
<html>
<head>        <title>Display functions in Javascript </title>
</head>
<body>
        <h1> Demo of console.log</h1> <br>
        <script>
            Console.log("Gurukul College");
            Console.log("7+5");
        </script>
</body>
</html>
```

Reserved Words:

Lists of all the reserved words in JavaScript are given in the following table. They cannot be used as Java Script variables, functions, methods, loop labels, or any object names.

Abstract	**Else**	**instance of**	**Switch**
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
Case	False	Native	Throws
Catch	Final	New	transient
Char	Finally	Null	True
Class	Float	Package	Try
Const	For	Private	Type of
Continue	Function	Protected	Var
Debugger	Go to	Public	Void
Default	If	Return	Volatile
Delete	Implements	Short	While
Do	Import	Static	With
Double	In	Super	

JavaScript Data Types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

JavaScript is a **dynamic type language**, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use **var** here to specify the data type. It can hold any type of values such as numbers, strings etc. For example:

```
var a=40;          //holding number
var b="Rahul";  //holding string
```

JavaScript primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	**Description**
1. String	represents sequence of characters e.g. var str="hello";
2. Number	represents numeric values e.g. var a=100;
3. Boolean	represents Boolean value either false or true
4. Undefined	represents undefined value. E.g. var address="Bharaptur" address= undefined
5. Null	represents null i.e. no value at all.

JavaScript non-primitive (reference) data types

The non-primitive data types are as follows:

Data Type	**Description**
1. Object	represents instance through which we can access members
2. Array	represents group of similar values
3. RegExp	represents regular expression

JavaScript Variable

1. JavaScript Local variable
2. JavaScript Global variable

A **JavaScript variable** is simply a name of storage location. There are two types of variables in JavaScript: local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore (_), or dollar($) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. Reserved words cannot be used as names.
4. JavaScript variables are case sensitive, for example x and X are different variables.

Correct JavaScript variables

1. var x = 10;
2. var _value="Hari";

Incorrect JavaScript variables

1. var 123=30;
2. var *aa=320;

Example of JavaScript variable

Let's see a simple example of JavaScript variable.

```
<html>
<head>
<title> Example of Variable </title>
<body>
<script>
var x = 10;
var y = 20;
var z=x+y;
document.write(z);
</script>
</body>
</html>
```

JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands. For example:

```
var sum=10+20;
```

Here, + is the arithmetic operator and = is the assignment operator.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators
7. Unary Operator
8. String Operator
9. Conditional Operator

JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
1) +	Addition	10+20 = 30
2) -	Subtraction	20-10 = 10
3) *	Multiplication	10*20 = 200
4) /	Division	20/10 = 2
5) %	Modulus (Remainder)	20%10 = 0
6) ++	Increment	var a=10; a++; Now a = 11
7)		

Examples

```
<html>
<body>
<h2>Arithmetic Operations</h2>
<p>A typical arithmetic operation takes two numbers (or expressions) and produces a new number.</p>
<p id="Hello"></p>
<script>
let a = 3;
let x = (100 + 50) * a;
document.getElementById("Hello").innerHTML=x;    // document.write(x); OR  window.alert(x);
</script>
</body>
</html>
```

JavaScript Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
1) ==	Is equal to	10==20 = false
2) !=	Not equal to	10!=20 = true
3) >	Greater than	20>10 = true
4) >=	Greater than or equal to	20>=10 = true
5) <	Less than	20<10 = false
6) <=	Less than or equal to	20<=10 = false

```
<html>
<body>
<h1>JavaScript Operators</h1>
<h2>The + Operator</h2>
<p>The + operator concatenates (adds) strings.</p>
<script>
let text1 = "Rajesh";
let text2 = "Hamal";
let text3 = text1 + " " + text2;
document.write(text3);
</script>
</body>
</html>
```

```
<html>
<body>
<h1>JavaScript Arithmetic</h1>
<h2>The += Operator</h2>
<script>
        var x = 10;
        x += 5;
document.write(x);
</script>
</body>
</html>
```

JavaScript Bitwise Operators

The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

Operator	Description	Example
1) &	Bitwise AND	(10==20 & 20==33) = false
2) |	Bitwise OR	(10==20 | 20==33) = false
3) ^	Bitwise XOR	(10==20 ^ 20==33) = false
4) ~	Bitwise NOT	(~10) = -10
5) <<	Bitwise Left Shift	(10<<2) = 40
6) >>	Bitwise Right Shift	(10>>2) = 2
7) >>>	Bitwise Right Shift with Zero	(10>>>2) = 2

JavaScript Logical Operators

The following operators are known as JavaScript logical operators.

Operator	Description	Example
1) &&	Logical AND	(10==20 && 20==33) = false
2) ||	Logical OR	(10==20 || 20==33) = false
3) !	Logical Not	!(10==20) = true

JavaScript Assignment Operators

The following operators are known as JavaScript assignment operators.

Operator	Description	Example
1) =	Assign	10+10 = 20
2) +=	Add and assign	var a=10; a+=20; Now a = 30
3) -=	Subtract and assign	var a=20; a-=10; Now a = 10
4) *=	Multiply and assign	var a=10; a*=20; Now a = 200
5) /=	Divide and assign	var a=10; a/=2; Now a = 5
6) %=	Modulus and assign	var a=10; a%=2; Now a = 0

JavaScript Special Operators

The following operators are known as JavaScript special operators.

Operator	Description
1) (?:)	Conditional Operator returns value based on the condition. It is like if-else.
2) ,	Comma Operator allows multiple expressions to be evaluated as single statement.
3) Delete	Delete Operator deletes a property from the object.
4) In	In Operator checks if object has the given property
5) Instanceof	checks if the object is an instance of given type
6) new	creates an instance (object)
7) typeof	checks the type of object.
8) Void	it discards the expression's return value.

JavaScript Bitwise Operators

Bit operators work on 32 bits numbers.

Any numeric operand in the operation is converted into a 32 bit number. The result is converted back to a JavaScript number.

Operator	Description	Example	Same as	Result	Decimal
&	AND	5 & 1	0101 & 0001	0001	1
|	OR	5 | 1	0101 | 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	left shift	5 << 1	0101 << 1	1010	10
>>	right shift	5 >> 1	0101 >> 1	0010	2

Functions and Function Return

A **JavaScript function** is a block of code designed to perform a particular task. A function is a go 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. Function allow a programmer to divide a big program into a number of small and manageable function.

Advantages of Using Function:

- You can't do anything in JavaScript without function.
- Functions increases 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 helps 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 di 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>
<title>
</title>
<head>
<script type="text/javascript">
function Hello()
  {
document.write ("Hello Class 12");
  }
</script>
</head>
<body>
    <p>Click the following button to call the function </p>
      <form>
            <input type="button" onclick="Hello()" value= "Click Here">
    </form>
</body>
</html>
```

Example 2

```
<html>
<body>
<script>
    function showMessage()
            {
    document.write ("Hello Class 12");
    document.write ("Good Morning");
            }
               showMessage()
</script>
</body>
</html>
```

Function Parameters

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

```
<html>
<body>
      <h2> Function with Parameters </h2>
<script>
      var a= 3
      var b= 4
           add(a,b);
              function add(a,b)
              {
              var sum=a+b;
document.write("sum of two number is="+sum);    // OR     alert("sum of two number is="+sum);
              }
</script>
</body>
<html>
```

Function Return

The **return** statement is used to return a particular value from the function to the function caller. The function will stop executing when the **return** statement is called. The **return** statement should be the last statement in a function because the code after the **return** statement will be unreachable.

```
<html>
<head>
<title> Example of Function Return</title>  </head>
<body>
   <h1> Example of the JavaScript's return statement </h1>
   <script>
      var a=12;
      var b= 20;
      var res = fun(a, b);
```

```
function fun(x, y)
{
var r= x*y;
return (r);
}
document.write(res);
   </script>
</body>
</html>
```

Above program is a simple example of using the **return** statement. Here, returning the result of the product of two numbers and returned the value to the function caller.
The variable **res** is the function caller; it calls the function **fun()** and passes two parameters as the arguments of the function.
The result will be store in the **res** variable. Final output 240 is the product of arguments **12** and **20**.

Control Structures

1. If-else
2. switch case
3. for loop
4. while loop
5. do while loop

The **JavaScript if-else statement** is used *to execute the code whether condition is true or false*. There are three forms of if statement in JavaScript.

1. If Statement
2. If else statement
3. if else if statement

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression)
{
//content to be evaluated
}
```

Let's see the simple example of if statement in JavaScript.

```
<script>
var a=20;
if(a>10)
{
document.write("value of a is greater than 10");
}
</script>
```

```
<html>
<head>
<title> Number to check Positive or not </title>
</head>
<body>
<script type= "text/javascript">
var num= prompt("Enter Number");
if (num>0)
{
        document.write("Given number is Positive");
}
</script>
</body>
</html>
```

JavaScript If...else Statement

It evaluates the content whether condition is true of false. The syntax of JavaScript if-else statement is given below.

```
if(expression)
{
//content to be evaluated if condition is true
}
else
{
//content to be evaluated if condition is false
}
```

```
<html>
<body>
<script>
var num= prompt("Enter Number");
if (num>0)
{
 document.write("Given number is Positive");
}
```

```
else
{
 document.write("Given number is Negative");
}
</script>
</body>
</html>
```

Let's see the example of if-else statement in JavaScript to find out the even or odd number.

```
<script>
var a=17;
if(a%2==0)
{
document.write("a is even number");
}
else
{
document.write("a is odd number");
}
</script>
```

```
<html>
<body>
<script>
var num= prompt("Enter Number");
var x=num%2;
if (x==0)
{
        document.write("Given number is Even");
}
else
{
        document.write("Given number is Odd");
}
</script>
</body>
</html>
```

JavaScript If...else if statement

It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

```
if(expression1)
{
//content to be evaluated if expression1 is true
}
else if(expression2)
{
//content to be evaluated if expression2 is true
}
else if(expression3)
{
//content to be evaluated if expression3 is true
}
else
{
//content to be evaluated if no expression is true
}
```

Let's see the simple example of if else if statement in JavaScript.

```
<script>
var a=20;
if(a==10)
{
document.write("a is equal to 10");
}
else if(a==15)
{
document.write("a is equal to 15");
}
else if(a==20)
{
document.write("a is equal to 20");
}
else
{
document.write("a is not equal to 10, 15 or 20");
}
</script>
```

Write a Program to input a number and check that is positive, negative or not.

```
<html>
<body>
<script>
var a=prompt("Enter Number");
if(a>0)
{
document.write("Number is positive");
}
else if(a<0)
{
document.write("Number is Negative");
}
else
{
document.write("Number is Zero");
}
</script>
</body>
</html>
```

JavaScript Switch

The **JavaScript switch statement** is used *to execute one code from multiple expressions*. It is just like else if statement that allow us to choose only one option among the many given options. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.

The signature of JavaScript switch statement is given below.

```
switch(expression)
{
case value1:
 code to be executed;
 break;
case value2:
 code to be executed;
 break;
.....
default:
 code to be executed if above values are not matched;
}
```

Let's see the simple example of switch statement in JavaScript.

```
<html>
<head>
<script>
var grade='B';
var result;
switch(grade)
{
case 'A':
result="A Grade";
break;
case 'B':
result="B Grade";
break;
case 'C':
result="C Grade";
break;
default:
result="No Grade";
}
document.write(result);
</script>
</head>
</html>
```

```
<html>
<head>
<title> Switch Case Example For to find Name of the day.</title>
</head>
<body>
<script>
	var n=prompt("Enter a number between 1 and 7");
	switch(n)
	{
		case(n="1");
		document.write("The day is Sunday");
		break;
		case(n="2");
		document.write("The day is Monday");
		break;
		case(n="3");
		document.write("The day is Tuesday");
```

```
break;
case(n="4");
document.write("The day is Wednesday");
break;
case(n="5");
document.write("The day is Thursday");
break;
case(n="6");
document.write("The day is Friday");
break;
default:
document.write("Invalid day");
break;
</script>
</body>
</html>
```

JavaScript Loops / Iterations

The processing of repeatedly executing the block of statements is called iteration or looping. Loops are used to repeat the execution of a block of code. The **JavaScript loops** are used *to iterate the piece of code* using for, while, do while. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop

1) JavaScript For loop

For loop is used to execute a set of statements for given number of times. The **JavaScript for loop** *iterates the elements for the fixed number of times*. It should be used if number of iteration is known. The syntax of for loop is given below.

```
for (initialization; condition; increment)
{
   code to be executed
}
```

Let's see the simple example of for loop in JavaScript.

```
<html>
<head>
<script>
for (i=1; i<=5; i++)
{
document.write(i + "<br/>")  // OR   document.write(i);
}
```

```
</script>
</head>
</html>
```

Output: 12345

JavaScript While loop:

The while loop is an entry controlled loop in which condition is checked before the execution of loop.

Syntax

```
while (condition)
{
  // code block to be executed
 }
```

Example, WAP to display even number from 1 to 20.

```
<script>
   var i=2;
     while (i<=20)
      {
        document.write(i);
         i=i+2;
       }
</script>
```

JavaScript do while loop

The **JavaScript do while loop** *iterates the elements for the infinite number of times* like while loop. But, code is *executed at least* once whether condition is true or false. The syntax of do while loop is given below.

```
do
{
   code to be executed
}while (condition);
```

Example,

```
<script>
var i=1;
do{
document.write(i + "<br/>");
i=i+2;
}while (i<=20);
</script>
```

Some JavaScript Program Examples

1. Write a JavaScript program to input two numbers and find out its sum.

```
<html>
<head>
<script type=”text/javascript”>
var a = prompt(“Enter the first number”);
var b = prompt(“Enter the second number”);
a = parseInt(a);
b = parseInt(b);
var sum=a+b
document.write(sum)
</head>
<body>
</body>
</html>
```

Or

```
<html>
<head>
        <title> To find out sum of Given two numbers </title>
<script type="text/JavaScript">
function add()
{
        var a,b,c;
a=Number(document.getElementaryById("first").value);
b=Number(document.getElementaryById("second").value);
c=a+b;
document.getElementaryById("answer").value=c;
}
</script>
</head>
<body>
        Enter the First Numebr:      <input id="first">       <br>
        Enter the Second Number:   <input id="second">   <br> <hr>
        <button onclick="add()"> Addition </button>
        <input id="answer">
</body>
</html>
```

Object based Programming with Java Script and Event Handling

Object

Object is a non-primitive data type in JavaScript. Object is like any other variable but it is quite differ from variable. The variable can hold only one value but object can hold multiple values. The object holds multiple values in terms of properties and methods. **Properties** can hold values of primitive data types and **methods** are functions.

An object is defined simple way as like a variable. The object is defined with a unique object name and the curly brackets { } is used to include properties and methods. The variables within the object is known as properties and the function within the object is known as methods. The properties or method are written as name: value pairs. The name and value separated by a colon. The period '.' is used to access value to object.

Syntax:

```
var object_name = { name: value1, name2: value2, nameN: valueN };
```

Example

```
<html>
<body>
<script>
    var emp={id:101,name:"Ram Prasad",salary:50000}
    document.write(emp.id+" "+emp.name+" "+emp.salary
    document.write(emp.id+" "+emp.name+" "+emp.salary);
</script>
</body>
</html>
```

In the above example, object name is defined as emp and it has three properties id, name and salary. The value of object can be accessed easily using object name period and properties name.

Creating Objects in JavaScript

There are 3 ways to create objects.

1. By object literal
2. By creating instance of Object directly (using new keyword)
3. By using an object constructor (using new keyword)

1) JavaScript Object by object literal

The syntax of creating object using object literal is given below:

```
object={property1:value1,property2:value2.... propertyN:valueN}
```

As you can see, property and value is separated by : (colon). Let's see the simple example of creating object in JavaScript.

```
<html>
<body>
<h2>JavaScript Objects</h2>
<p>Creating a JavaScript Object:</p>
<p id="demo"></p>
<script>
const person = {firstName: "John", lastName: "Doe", age: 50, eyeColor: "blue"};
document.getElementById("demo").innerHTML = person.firstName + " is " + person.age + " years old.";
</script>
</body>
</html>
```

```
<html>
<head>
< title></title>
< script>
    var vehicle = {
                    color:"green",
                    weight:50,
                    height:5,
                    move:function()
                      {
                      document.write("vehicle moves");
                      }
                  };
      vehicle.move();
  </script>
</head>
<body>
</body>
</html>
```

- *In the above example, method is defined as move. The method can be accessed easily using object name period and method name.*

2) By creating instance of Object

The syntax of creating object directly is given below:

```
var objectname=new Object();
```

Here, **new keyword** is used to create object. Let's see the example of creating object directly.

```
<html>
<body>
<script>
var study=new Object();
study.roll=10;
study.name="Rajesh Hamal";
study.class=12;
document.write(study.roll +"<br>");
document.write(study.name +"<br>");
document.write(study.class);
</script>
</body>
</html>
```

3) By using an Object constructor

Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.

The **this keyword** refers to the current object.

The example of creating object by object constructor is given below.

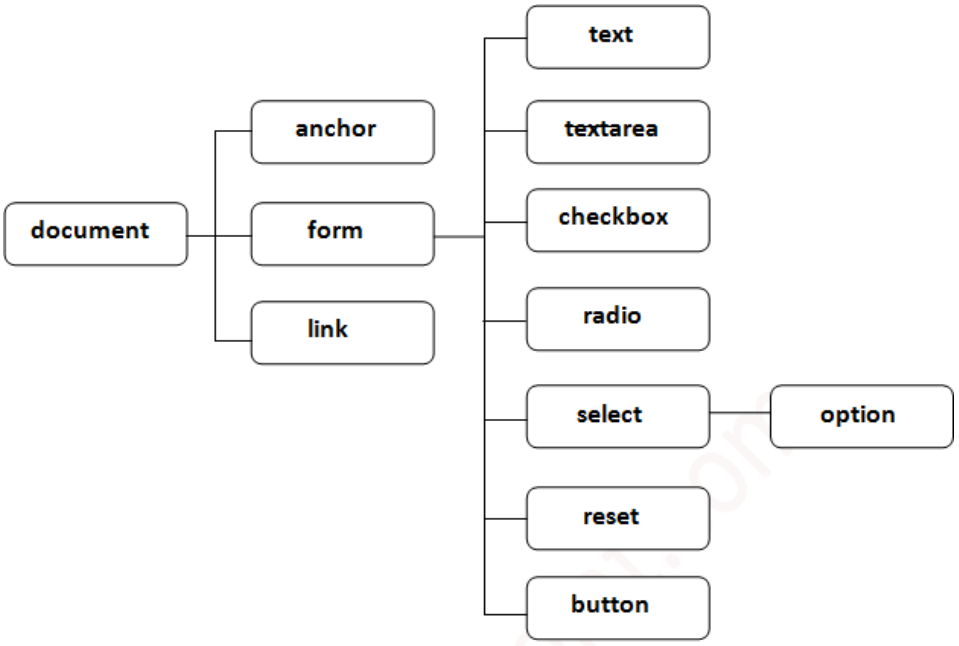
```
<script>
function emp(id,name,salary)
{
this.id=id;
this.name=name;
this.salary=salary;
}
e=new emp(103,"Anjan Bista",30000);
document.write(e.id+" "+e.name+" "+e.salary);
</script
```

Document Object Model

The **document object** represents the whole html document. When html document is loaded in the browser, it becomes a document object. It is the **root element** that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page. As mentioned earlier, it is the object of window. So **window.document** is same as a **document**

Properties of document object

Let's see the properties of document object that can be accessed and modified by the document object.



JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles on the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

Methods of document object

We can access and change the contents of document by its methods. The important methods of document object are as follows:

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

Accessing field value by document object

In this example, we are going to get the value of input text by user. Here, we are using **document.form1.name.value** to get the value of name field. Here, **document** is the root element that represents the html document. **form1** is the name of the form. **name** is the attribute name of the input text. **value** is the property, that returns the value of the input text.

Let's see the simple example of document object that prints name with welcome message.

```
<script type="text/javascript">
function printvalue()
{
var name=document.form1.name.value;
alert("Welcome: "+name);
}
</script>
  <form name="form1">
     Enter Name:<input type="text" name="name"/>
<input type="button" onclick="printvalue()" value="print name"/>
</form>
```

1. getElementById( )

The **document.getElementById()** method returns the element of specified id. We can use document.getElementById() method to get value of the input text. But we need to define id for the input field.

Example

```
<html>
<body>
<p id="demo"></p>
<p id="try"></p>
<script>
var x="How are you"
document.getElementById("demo").innerHTML = x;
document.getElementById("try").innerHTML= " I am fine and you"
</script>
```

*In the example above, getElementById is a **method**, while innerHTML is a **property**.*

Let's see the simple example of **document.getElementById()** method that prints cube of the given number.

```
<script>
function getcube()
{
var n=document.getElementById("number").value;
alert(n*n*n);
}
</script>
<form>
Enter No:<input type="text" id="number" name="number"/><br/>
<input type="button" value="cube" onclick="getcube()"/>
</form>
```

2. getElementsByName()

The **document.getElementsByName()** method returns all the element of specified name. The syntax of the getElementsByName() method is given below:

`document.getElementsByName("name")` Here, name is required.

Example of document.getElementsByName() method

In this example, we going to count total number of genders. Here, we are using getElementsByName() method to get all the genders.

```
<script>
function totalelements()
{
var g=document.getElementsByName("gender");
alert("Total Genders:"+g.length);
}
</script>
<form>
Male:<input type="radio" name="gender" value="male">
Female:<input type="radio" name="gender" value="female">

<input type="button" onclick="totalelements()" value="Total Genders">
</form>
```

3. document.getElementsByTagName()

The **document.getElementsByTagName()** method returns all the element of specified tag name.

The syntax of the getElementsByTagName() method is given below:

```
document.getElementsByTagName("name")
```

Here, name is required.

Example of document.getElementsByTagName() method

In this example, we going to count total number of paragraphs used in the document. To do this, we have called the document.getElementsByTagName("p") method that returns the total paragraphs.

```
<script type="text/javascript">
function countpara()
{
var totalpara=document.getElementsByTagName("p");
alert("total p tags are: "+totalpara.length);
}
</script>
<p>This is a pragraph</p>
<p>Here we are going to count total number of paragraphs by getElementByTagName() method.
</p>
<p>Let's see the simple example</p>
<button onclick="countpara()">count paragraph</button>
```

Another example of document.getElementsByTagName() method

In this example, we going to count total number of h2 and h3 tags used in the document.

```
<script type="text/javascript">
function counth2()
{
var totalh2=document.getElementsByTagName("h2");
alert("total h2 tags are: "+totalh2.length);
}
function counth3()
{
var totalh3=document.getElementsByTagName("h3");
alert("total h3 tags are: "+totalh3.length);
}
</script>
<h2>This is h2 tag</h2>
<h2>This is h2 tag</h2>
<h3>This is h3 tag</h3>
<h3>This is h3 tag</h3>
<h3>This is h3 tag</h3>
<button onclick="counth2()">count h2</button>
<button onclick="counth3()">count h3</button>
```

Javascript - innerHTML

The **innerHTML** property can be used to write the dynamic html on the html document.

It is used mostly in the web pages to generate the dynamic html such as registration form, comment form, links etc.

Example of innerHTML property

In this example, we are going to create the html form when user clicks on the button.

In this example, we are dynamically writing the html form inside the div name having the id mylocation. We are identifying this position by calling the document.getElementById() method.

```
<script type="text/javascript" >
function showcommentform()
 {
var data="Name:<input type='text' name='name'><br>
Comment:<br><textarea rows='5' cols='80'></textarea>
<br><input type='submit' value='Post Comment'>";
document.getElementById('mylocation').innerHTML=data;
}
</script>
<form name="myForm">
<input type="button" value="comment" onclick="showcommentform()">
<div id="mylocation"></div>  </form>
```

JavaScript Events

The change in the state of an object is known as an **Event**. In html, there are various events which represents that some activity is performed by the user or by the browser. When JavaScript code is included in HTML, JavaScript react over these events and allow the execution. This process of reacting over the events is called **Event Handling**. Thus, JavaScript handles the HTML events via **Event Handlers**.

For example, when a user clicks over the browser, add js code, which will execute the task to be performed on the event.

Some of the HTML events and their event handlers are:

Mouse events:

Event Performed	Event Handler	Description
Click	Onclick	When mouse click on an element
Mouseover	Onmouseover	When the cursor of the mouse comes over the element
Mouseout	Onmouseout	When the cursor of the mouse leaves an element
Mousedown	Onmousedown	When the mouse button is pressed over the element
Mouseup	Onmouseup	When the mouse button is released over the element
Mousemove	Onmousemove	When the mouse movement takes place.

Keyboard events:

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

Form events:

Event Performed	Event Handler	Description
Focus	onfocus	When the user focuses on an element
Submit	onsubmit	When the user submits the form
Blur	Onblur	When the focus is away from a form element
Change	onchange	When the user modifies or changes the value of a form element

Window/Document events

Event Performed	Event Handler	Description
Load	Onload	When the browser finishes the loading of the page
Unload	onunload	When the visitor leaves the current webpage, the browser unloads it
Resize	Onresize	When the visitor resizes the window of the browser

Let's discuss some examples over events and their handlers.

Click Event

```
<html>
<head> Javascript Events </head>
<body>
<script>
      function clickevent()
          {
              document.write("This is Java Click Events");
          }
</script>
        <form>
                    <input type="button" onclick="clickevent()" value="Click To View"/>
        </form>
```

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

Mouse Over Event

```
<html>
<head>
<h1> Javascript Events </h1>
</head>
<body>
<script>
        function mouseoverevent()
    {
        alert("This is Java MouseOverEvent");
    }
   </script>
<p onmouseover="mouseoverevent()"> Keep Mouse's cursor over me</p>
</body>
</html>
```

Focus Event

```
<html>
<head> Javascript Events</head>
<body>
<h2> Enter something here</h2>
<input type="text" id="input1" onfocus="focusevent()"/>
<script>
    function focusevent()
    {
        document.getElementById("input1").style.background=" aqua";
    }
</script>
</body>
</html>
```

Keydown Event

```
<html>
<head> Javascript Events</head>
<body>
<h2> Enter something here</h2>
<input type="text" id="input1" onkeydown="keydownevent()"/>
<script>
    function keydownevent()
    {
        document.getElementById("input1");
        alert("Pressed a key");
    }
</script>
</body>
</html>
```

Load event

```
<html>
<head>Javascript Events</head>
```

```
</br>
<body onload="window.alert('Page successfully loaded');">
<script>
   document.write("The page is loaded successfully");
</script>
</body>
</html>
```

Image Object

Form Validation

It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

In this example, we are going to validate the name and password. The name can’t be empty and password can’t be less than 6 characters long.

Here, we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

```
<html>
<body>
<script>
function validateform()
{
var name=document.myform.name.value;
var password=document.myform.password.value;

if (name==null || name=="")
{
  alert("Name can't be blank");
  return false;
}
else if(password.length<6)
{
  alert("Password must be at least 6 characters long.");
  return false;
  }
}
</script>
<body>
```

```
<form name="myform" method="post"
action="http://www.javatpoint.com/javascriptpages/valid.jsp"
onsubmit="return validateform()" >
Name: <input type="text" name="name"><br/>
Password: <input type="password" name="password"><br/>
<input type="submit" value="register">
</form>
</body>
</html>
```

JavaScript Retype Password Validation

```
<script type="text/javascript">
function matchpass(){
var firstpassword=document.f1.password.value;
var secondpassword=document.f1.password2.value;

if(firstpassword==secondpassword){
return true;
}
else{
alert("password must be same!");
return false;
}
}
</script>

<form name="f1" action="register.jsp" onsubmit="return matchpass()">
Password:<input type="password" name="password" /><br/>
Re-enter Password:<input type="password" name="password2"/><br/>
<input type="submit">
</form>
```

Number Validation

Let's validate the textfield for numeric value only. Here, we are using isNaN() function.

```
<script>
function validate()
{
var num=document.myform.num.value;
if (isNaN(num))
{
  document.getElementById("numloc").innerHTML="Enter Numeric value only";
```

```
  return false;
}
Else
{
  return true;
  }
}
</script>
<form name="myform" onsubmit="return validate()" >
Number: <input type="text" name="num"><span id="numloc"></span><br/>
<input type="submit" value="submit">
</form>
```

JavaScript validation with image

Let's see an interactive JavaScript form validation example that displays correct and incorrect image if input is correct or incorrect.

```
<script>
function validate()
{
var name=document.f1.name.value;
var password=document.f1.password.value;
var status=false;

if(name.length<1)
{
document.getElementById("nameloc").innerHTML=
" <img src='unchecked.gif'/> Please enter your name";
status=false;
}
Else
{
document.getElementById("nameloc").innerHTML=" <img src='checked.gif'/>";
status=true;
}
if(password.length<6)
{
document.getElementById("passwordloc").innerHTML=
" <img src='unchecked.gif'/> Password must be at least 6 char long";
status=false;
```

```
}
Else
{
document.getElementById("passwordloc").innerHTML=" <img src='checked.gif'/>";

}
return status;
}
</script>
<form name="f1" action="#" onsubmit="return validate()">
<table>
<tr><td>Enter Name:</td><td><input type="text" name="name"/>
<span id="nameloc"></span></td></tr>
<tr><td>Enter Password:</td><td><input type="password" name="password"
/>
<span id="passwordloc"></span></td></tr>
<tr><td colspan="2"><input type="submit" value="register"/></td></tr>
</table>
</form>
```

JavaScript email validation

We can validate the email by the help of JavaScript.

There are many criteria that need to be follow to validate the email id such as:

- Email id must contain the @ and . character
- There must be at least one character before and after the @.
- There must be at least two characters after. (Dot).

Let's see the simple example to validate the email field.

```
<script>
function validateemail()
{
var x=document.myform.email.value;
var atposition=x.indexOf("@");
var dotposition=x.lastIndexOf(".");
if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length)
{
  alert("Please enter a valid e-
mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
```

```
  return false;
  }
}
</script>
<body>
<form name="myform"  method="post" action="#" onsubmit="return validateemail
();">
Email: <input type="text" name="email"><br/>
  <input type="submit" value="register">
</form>
```

JavaScript Programs

1. **This program adds 5 numbers and writes the answer as an HTML output.**

```
<!doctype html>
<html> <body>
<script>
var i, n=5, sum=0; arr = new Array(1, 2, 3, 5, 9);
for(i=0; i<5; i++)
sum = sum + arr[i]; document.write(sum);
</script></body></html>
```

2. **JavaScript program to check whether a number is a prime number or not. This program does not take input from user.**

```
<!doctype html>
<html>
<body> <script>
var num, i, c=0;
num=9;
for(i=2; i<num; i++)
{
if(num%2==0)
{
C++;
break;
}
}
if(c==0)
document.write(num+" is a Prime Number");
else
document.write(num+" is not a Prime Number");
</script> </body>
</html>
```

Data Format Validation:

Secondly, the data that is entered must be checked for correct form and value. Your code must include

appropriate logic to test the correctness of data.

```
<html>
<body>
<script>
function myFormValidate ( )
{
var name = document.myform.name.value; var password =
document.myform.password.value;
    if (name==null || name=="")
    {
    alert("Please, Enter Name");
    return false;
}
else if (password.length < 5)
{
 alert ("Password must be at least 5 characters long.");
return false;
    }
}
</script>
<body>
<form name = "myform" method = "post" action = "abc.php" onsubmit = "return myFormValidate ( )" >
Name: <input type = "text" name = "name"><br/>
Password: <input type = "password" name = "password"><br/>
<input type = "submit" value = "Login">
</form>
</body>
</html>
```

JQuery

JQuery is a fast, small, lightweight, "write less, do more", and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling animation, and Ajax much simpler with an easy-to-use API (Application Programming Interface) that works across browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

JQuery is a JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. JQuery simplifies a lot of the complicated things from JavaScript like AJAX (Asynchronous JavaScript and XML) calls and DOM (Document Object Model) manipulation. Some of the biggest companies which uses jQuery on the Web are

1. Google 2. Microsoft 3. IBM

Features of jQuery:

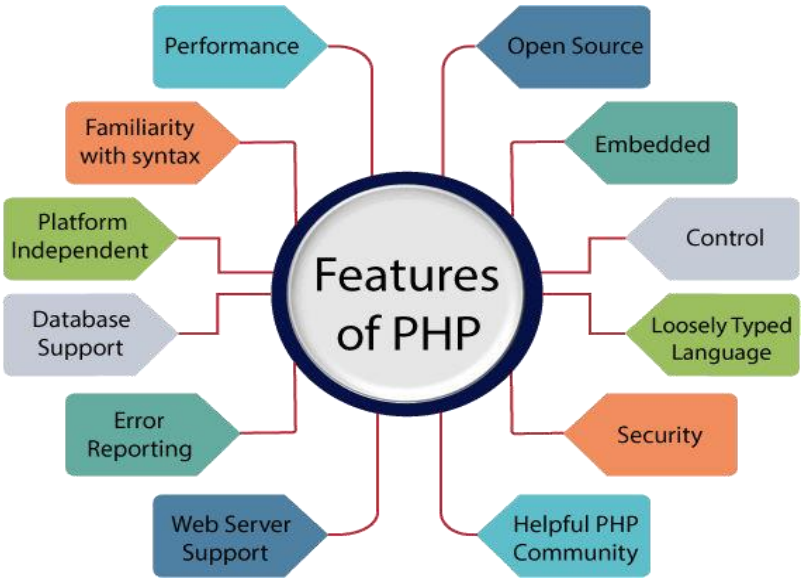
Some of important feature of jQuery are listed as follows:

1. The jQuery is very small, fast, lightweight JavaScript library
2. It is very fast and extensible.
3. It has a lot of built-in animation effects which can use directly in websites.
4. It is used to improve the performance of an application.
5. It is used to develop browser's compatible web applications effectively.
6. It uses mostly new features of new browsers.
7. It is platform-independent.

8. It is used to develop a responsive web application.

Server Side Scripting Using PHP:

Server side script are discrete blocks of program code that execute on a web server (as opposed to a web client) computer. They are generally used to create Dynamic web pages. This means that the page displayed to the user does not exist as a document on the server in its own right, and is only brought into existence in response to a user request. Often, a server-side script provides the interface between a web-based user interface and a database that resides on a web server. PHP is a server side scripting language used to create dynamic web pages. PHP is well suited for web development. PHP stands for hypertext preprocessor. It is an interpreted language. It is embedded in HTML. It is an open source language.



Some important points need to be notice about PHP are as followed:

1. PHP stands for Hypertext Preprocessor.
2. PHP is an interpreted language, i.e., there is no need for compilation.
3. PHP is faster than other scripting languages, for example, ASP and JSP.
4. PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
5. PHP can be embedded into HTML.
6. PHP is an object-oriented language.
7. PHP is an open-source scripting language.
8. PHP is simple and easy to learn language.

Features of PHP

1. Simple
2. Interpreted
3. Faster
4. Open Source
5. Platform Independent
6. Case Sensitive
7. Error Reporting
8. Real-Time Access Monitoring
9. Loosely Typed Language

Characteristics of PHP

1. Simplicity
2. Efficiency
3. Security
4. Flexibility
5. Familiarity

Advantages of PHP / what problem does it solve.

1. PHP is Free
2. PHP is Cross Platform
3. PHP is widely used
4. PHP hides its complexity
5. PHP is built for Web Programming

Introduction to PHP:

PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).

Before learning PHP, you must have the basic knowledge of **HTML, CSS,** and **JavaScript**. So, learn these technologies for better implementation of PHP.

HTML - HTML is used to design static webpage.

CSS - CSS helps to make the webpage content more effective and attractive.

JavaScript - JavaScript is used to design an interactive website.

PHP is widely used in web development nowadays. PHP can develop dynamic websites easily. But you must have the basic the knowledge of following technologies for web development as well.

1. HTML
2. CSS
3. JavaScript
4. Ajax
5. XML and JSON
6. jQuery

Why use PHP / Strengths of PHP

PHP is a server-side scripting language, which is used to design the dynamic web applications with MySQL database.

1. It handles dynamic content, database as well as session tracking for the website.
2. You can create sessions in PHP.
3. It can access cookies variable and also set cookies.
4. It helps to encrypt the data and apply validation.
5. PHP supports several protocols such as HTTP (Hypertext Transfer Protocol), POP3 (Post office Protocol-3), SNMP (Simple Network Management Protocol), LDAP (Lightweight Directory Access Protocol), IMAP (Internet Message Access Protocol), and many more.
6. Using PHP language, you can control the user to access some pages of your website.
7. As PHP is easy to install and set up, this is the main reason why PHP is the best language to learn.
8. PHP can handle the forms, such as - collect the data from users using forms, save it into the database, and return useful information to the user. **For example** - Registration form.

Weaknesses of PHP

1. PHP's main strength flexibility is also its weakness. It can be a little too forgiving of errors.
2. With no strict rules, inexperienced programmers have the freedom to create some very bad solutions to simple problems.
3. Bad packages that got popular in the community are still reused when developers are trying something new or rushed for time. Some of these mistakes lead to security risks.

4. As a mature tool PHP has some legacy baggage. There are lots of internal consistencies, especially surrounding references and values.
5. This is mostly due to updates, which add features that clash with earlier features,
6. PHP is an interpreted language, which can reduce speed.
7. PHP 7 increased performance over previous versions by a significant amount while maintaining most language compatibility.
8. The changes didn't affect the learning curve or existing applications much while improving performance. Still, it executes more slowly than compiled languages.
9. Scaling and maintaining PHP is a complicated endeavor.
10. Context matters a great deal in dynamically typed languages, so tracking down errors gets harder the larger an application grows.
11. Experienced PHP developers can mitigate this problem y planning for scalability, but there's only so much they can do to reduce maintenance issues down the road.

Common uses of PHP

1. PHP performs system functions, i.e., from files on a system it can create, open, read, write, and close them.
2. PHP can handle forms, i.e., gather data from files, save data to a file, through email we can send data, return data to the user.
3. We add, delete, and modify elements within the database through PHP.
4. Access cookies variables and set cookies.
5. Using PHP, we can restrict users to access some pages of your website.
6. It can encrypt data.

3.15 Object Oriented Programming with Server Side Scripting

Object Oriented is an approach to software development that models application around real world objects such as employees, cars, bank accounts, etc. A class defines the properties and methods of a real world object. An object is an occurrence of a class. There are three major principles of OOP are:

1. Encapsulation: This is concerned with hiding the implementation details and only exposing the methods. It used to reduce software development complexity.

2. Inheritance: This is concerned with the relationship between classes. The main purpose of inheritance is the re-usability.

3. Polymorphism: This is concerned with having a single form but many different implementation ways. The main purpose of polymorphism is to simplify maintaining applications and making them more extendable.

PHP in Object oriented programming PHP is an object oriented scripting language. It supports all of the above principles. The above principles are achieved via:

Encapsulation - using "get" and "set" methods.

Inheritance - using extends keyword.

Polymorphism - using implements keyword

Hardware and Software Requirements

Hardware

Hardware requirement are not much, we just need a laptop or desktop computer. PHP 5.5+ require a computer with at least window 2008/vista.
The computer with Pentium IV processor, RAM 2GB, hard disk 256 GB, color monitor or above is more than enough.

Software

To run PHP code, we need the following software on our local machine.

1. Browser
2. Web server (eg. Apache)
3. PHP (interpreter)
4. MySQL Database (it is optional)

Basics of PHP:

Setting up the environment

To run PHP a web development is needed. This needs a PHP compatible web server and interpreter. Package like WAMP, LAMP, and XAMP etc. can be used which includes a web server.

Writing the code and running the script

PHP scripts are plain text. A PHP script begins with <?php and ends with ?>. The PHP code is saved with extension. PHP and is saved in the root directory of web server.

Basic PHP Syntax

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.
- Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

```
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello Class 12";
?>

</body>
</html>
```

Note: PHP statements end with a semicolon (;).

PHP Case Sensitivity

In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.

In the example below, all three echo statements below are equal and legal:

```
<html>
<body>

<?php
ECHO "Hello Class 12<br>";
echo "Hello Class 12<br>";
EcHo "Hello Class 12<br>";
?>

</body>
</html>
```

Comments in PHP

Comments are used to make code more readable. There are two types of comments –single line and multi-line comments. A single line comments starts with // while multi-line comment begins with /* and end with */.

```
<html>
<body>

<?php
// This is a single-line comment

# This is also a single-line comment
?>

</body>
</html>
```

```
<html>
 <body>

 <?php
 /*
 This is a multiple-lines comment block
 that spans over multiple
 lines
 */
 ?>

 </body>
 </html>
```

Loosely typed language: PHP is a loosely typed language, it means PHP automatically converts the variable to its correct data type.

PHP Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

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)

PHP Variable: Declaring string, integer, and float

Let's see the example to store string, integer, and float values in PHP variables.

```
<?php
$str="hello string";
$x=200;
$y=44.6;
echo "string is: $str <br/>";
echo "integer is: $x <br/>";
echo "float is: $y <br/>";
?>
```

PHP Variable Scope

The scope of a variable is defined as its range in the program under which it can be accessed. In other words, "The scope of a variable is the portion of the program within which it is defined and can be accessed."

PHP has three types of variable scopes:

1. Local variable 2. Global variable 3. Static variable

Local variable

The variables that are declared within a function are called local variables for that function. These local variables have their scope only in that particular function in which they are declared. This means that these variables cannot be accessed outside the function, as they have local scope.

A variable declaration outside the function with the same name is completely different from the variable declared inside the function. Let's understand the local variables with the help of an example:

Global variable

The global variables are the variables that are declared outside the function. These variables can be accessed anywhere in the program. To access the global variable within a function, use the GLOBAL keyword before the variable. However, these variables can be directly accessed or used outside the function without any keyword. Therefore there is no need to use any keyword to access a global variable outside the function.

Let's understand the global variables with the help of an example:

```
<?php
   function local_var()
   {
      $num = 45; //local variable
      echo "Local variable declared inside the function is: ". $num;
   }
   local_var();
?>
```

```
<?php
$name = "Sanaya Sharma";        //Global Variable
function global_var()
{
    global $name;
    echo "Variable inside the function: ". $name;
    echo "</br>";
}
global_var();
echo "Variable outside the function: ". $name;
?>
```

Basic Programming in PHP

PHP echo and print Statements

We frequently use the echo statement to display the output. There are two basic ways to get the output in PHP:

- echo
- print

echo and print are language constructs, and they never behave like a function. Therefore, there is no requirement for parentheses. However, both the statements can be used with or without parentheses. We can use these statements to output variables or strings.

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters."; <br> <hr>

print "<h2>PHP is Case Sensitive!</h2>";
print "Hello Class 12!<br>";
print "I'm about to learn PHP!";
?>
```

Difference between echo and print

echo

- echo is a statement, which is used to display the output.
- echo can be used with or without parentheses.
- echo does not return any value.
- We can pass multiple strings separated by comma (,) in echo.
- echo is faster than print statement.

print

- print is also a statement, used as an alternative to echo at many times to display the output.
- print can be used with or without parentheses.

- print always returns an integer value, which is 1.

- Using print, we cannot pass multiple arguments.
- print is slower than echo statement.

PHP Data Types

Variables can store data of different types, and different data types can do different things. PHP supports the following data types:

1. String
2. Integer
3. Float (floating point numbers - also called double)
4. Boolean
5. Array
6. Object
7. NULL
8. Resource

PHP Data Types: Compound Types

It can hold multiple values. There are 2 compound data types in PHP.

1. array
2. object

PHP Data Types: Special Types

There are 2 special data types in PHP.

1. resource
2. NULL

PHP Boolean

Booleans are the simplest data type works like switch. It holds only two values: **TRUE (1)** or **FALSE (0)**. It is often used with conditional statements. If the condition is correct, it returns TRUE otherwise FALSE.

Example:

```
<?php
   if (TRUE)
      echo "This condition is TRUE.";
   if (FALSE)
      echo "This condition is FALSE.";
?>
```

Output:

```
This condition is TRUE.
```

PHP Integer

Integer means numeric data with a negative or positive sign. It holds only whole numbers, i.e., numbers without fractional part or decimal points.

Rules for integer:

- An integer can be either positive or negative.
- An integer must not contain decimal point.
- Integer can be decimal (base 10), octal (base 8), or hexadecimal (base 16).
- The range of an integer must be lie between 2,147,483,648 and 2,147,483,647 i.e., -2^31 to 2^31.

Example:

```
<?php
   $dec1 = 34;
   $oct1 = 0243;
   $hexa1 = 0x45;
   echo "Decimal number: " .$dec1. "</br>";
   echo "Octal number: " .$oct1. "</br>";
   echo "HexaDecimal number: " .$hexa1. "</br>";
?>
```

Output:

```
Decimal number: 34
Octal number: 163
Hexadecimal number: 69
```

PHP Float

A floating-point number is a number with a decimal point. Unlike integer, it can hold numbers with a fractional or decimal point, including a negative or positive sign.

Example:

```
<?php
   $n1 = 19.34;
   $n2 = 54.472;
   $sum = $n1 + $n2;
   echo "Addition of floating numbers: " .$sum;
?>
```

Output:

```
Addition of floating numbers: 73.812
```

PHP String

A string is a non-numeric data type. It holds letters or any alphabets, numbers, and even special characters. String values must be enclosed either within **single quotes** or in **double quotes**. But both are treated differently. To clarify this, see the example below:

Example:

```
<?php
   $company = "Javatpoint";
   //both single and double quote statements will treat different
   echo "Hello $company";
   echo "</br>";
   echo 'Hello $company';
?>
```

Output:

```
Hello Javatpoint
Hello $company
```

PHP Array

An array is a compound data type. It can store multiple values of same data type in a single variable.

Example:

```
<?php
    $bikes = array ("Royal Enfield", "Yamaha", "KTM");
    var_dump($bikes); //the var_dump() function returns the datatype and values
    echo "</br>";
    echo "Array Element1: $bikes[0] </br>";
    echo "Array Element2: $bikes[1] </br>";
    echo "Array Element3: $bikes[2] </br>";
?>
```

Output:

```
array(3) { [0]=> string(13) "Royal Enfield" [1]=> string(6) "Yamaha" [2]=> string(3) "KTM" }
Array Element1: Royal Enfield
Array Element2: Yamaha
Array Element3: KTM
```

PHP object

Objects are the instances of user-defined classes that can store both values and functions. They must be explicitly declared.

Example:

```
<?php
    class bike {
        function model() {
            $model_name = "Royal Enfield";
            echo "Bike Model: " .$model_name;
          }
    }
    $obj = new bike();
    $obj -> model();
?>
```

Output:

```
Bike Model: Royal Enfield
```

PHP Resource

Resources are not the exact data type in PHP. Basically, these are used to store some function calls or references to external PHP resources. **For example** - a database call. It is an external resource.
This is an advanced topic of PHP, so we will discuss it later in detail with examples.

PHP Null

Null is a special data type that has only one value: **NULL**. There is a convention of writing it in capital letters as it is case sensitive.
The special type of data type NULL defined a variable with no value.

Example:

```
<?php
   $n = NULL;
```

```
   echo $n;  //it will not give any output
?>
```

PHP Operators

PHP Operator is a symbol i.e used to perform operations on operands. In simple words, operators are used to perform operations on variables or values. For example:

$num=10+20; //+ is the operator and 10,20 are operands

In the above example, + is the binary + operator, 10 and 20 are operands and $num is variable.

PHP Operators can be categorized in following forms:

1. Arithmetic Operators
2. Assignment Operators
3. Bitwise Operators
4. Comparison Operators
5. Incrementing/Decrementing Operators
6. Logical Operators
7. String Operators
8. Array Operators
9. Type Operators
10. Execution Operators

Arithmetic Operators

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

Operator	Name	Example	Explanation
+	Addition	$a + $b	Sum of operands
-	Subtraction	$a - $b	Difference of operands
*	Multiplication	$a * $b	Product of operands
/	Division	$a / $b	Quotient of operands
%	Modulus	$a % $b	Remainder of operands
**	Exponentiation	$a ** $b	$a raised to the power $b

Assignment Operators

The assignment operators are used to assign value to different variables. The basic assignment operator is "=".

Operator	Name	Example	Explanation
=	Assign	$a = $b	The value of right operand is assigned to the left operand.
+=	Add then Assign	$a += $b	Addition same as $a = $a + $b
-=	Subtract then Assign	$a -= $b	Subtraction same as $a = $a - $b
*=	Multiply then Assign	$a *= $b	Multiplication same as $a = $a * $b
/=	Divide then Assign (quotient)	$a /= $b	Find quotient same as $a = $a / $b

%=	Divide then Assign (remainder)	$a %= $b	Find remainder same as $a = $a % $b

Bitwise Operators

The bitwise operators are used to perform bit-level operations on operands. These operators allow the evaluation and manipulation of specific bits within the integer.

Operator	Name	Example	Explanation
&	And	$a & $b	Bits that are 1 in both $a and $b are set to 1, otherwise 0.
|	Or (Inclusive or)	$a | $b	Bits that are 1 in either $a or $b are set to 1
^	Xor (Exclusive or)	$a ^ $b	Bits that are 1 in either $a or $b are set to 0.
~	Not	~$a	Bits that are 1 set to 0 and bits that are 0 are set to 1
<<	Shift left	$a << $b	Left shift the bits of operand $a $b steps
>>	Shift right	$a >> $b	Right shift the bits of $a operand by $b number of places

Comparison Operators

Comparison operators allow comparing two values, such as number or string. Below the list of comparison operators are given:

Operator	Name	Example	Explanation
==	Equal	$a == $b	Return TRUE if $a is equal to $b
===	Identical	$a === $b	Return TRUE if $a is equal to $b, and they are of same data type
!==	Not identical	$a !== $b	Return TRUE if $a is not equal to $b, and they are not of same data type
!=	Not equal	$a != $b	Return TRUE if $a is not equal to $b
<>	Not equal	$a <> $b	Return TRUE if $a is not equal to $b
<	Less than	$a < $b	Return TRUE if $a is less than $b
>	Greater than	$a > $b	Return TRUE if $a is greater than $b
<=	Less than or equal to	$a <= $b	Return TRUE if $a is less than or equal $b
>=	Greater than or equal to	$a >= $b	Return TRUE if $a is greater than or equal $b
<=>	Spaceship	$a <=>$b	Return -1 if $a is less than $b Return 0 if $a is equal $b Return 1 if $a is greater than $b

Incrementing/Decrementing Operators

The increment and decrement operators are used to increase and decrease the value of a variable.

Operator	Name	Example	Explanation
++	Increment	++$a	Increment the value of $a by one, then return $a
		$a++	Return $a, then increment the value of $a by one
--	Decrement	--$a	Decrement the value of $a by one, then return $a
		$a--	Return $a, then decrement the value of $a by one

Logical Operators

The logical operators are used to perform bit-level operations on operands. These operators allow the evaluation and manipulation of specific bits within the integer.

Operator	Name	Example	Explanation
And	And	$a and $b	Return TRUE if both $a and $b are true
Or	Or	$a or $b	Return TRUE if either $a or $b is true
Xor	Xor	$a xor $b	Return TRUE if either $ or $b is true but not both
!	Not	! $a	Return TRUE if $a is not true
&&	And	$a && $b	Return TRUE if either $a and $b are true
||	Or	$a || $b	Return TRUE if either $a or $b is true

PHP Form Handling

We can create and use forms in PHP. To get form data, we need to use PHP super global $_GET and $_POST. The form request may be get or post. To retrieve data from get request, we need to use $_GET, for post request $_POST.

PHP Get Form

Get request is the default form request. The data passed through get request is visible on the URL browser so it is not secured. You can send limited amount of data through get request.

Let's see a simple example to receive data from get request in PHP.

File: form1.html

```
<form action="welcome.php" method="get">
Name: <input type="text" name="name"/>
<input type="submit" value="visit"/>
</form>
```

File: welcome.php

```
<?php
$name=$_GET["name"];//receiving name field value in $name variable
echo "Welcome, $name";
?>
```

PHP Post Form

Post request is widely used to submit form that have large amount of data such as file upload, image upload, login form, registration form etc.

The data passed through post request is not visible on the URL browser so it is secured. You can send large amount of data through post request.

Let's see a simple example to receive data from post request in PHP.

File: form1.html

```
<html>
<head> <title>  Post Method </title>
</head>
```

```
<body>
<form action="login.php" method="post">
<table>
<tr><td>Name:</td><td> <input type="text" name="name"/></td></tr>
<tr><td>Password:</td><td> <input type="password" name="password"/></td></tr>
<tr><td colspan="2"><input type="submit" value="login"/>  </td></tr>
</table>
</form> </body> </html>
```

File: login.php

```
<?php
$name=$_POST["name"];
$password=$_POST["password"];
  echo "Welcome: $name, your password is: $password";
?>
```

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

PHP Programs

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>
```

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

```
<html>
<head>
<title> Your name </title>
</head>
<body>
        <B> Factorial Calculaiton</B> <br/>
<form 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>
```

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

```
<html>
        <head>
        <title> Your name </title>
        </head>
        <body>
                <B> Display Even number up to 50</B> <br/> <hr>
        <?php
                $a=2;
                for($i=1;$i<=25;$i++)
                {
                        echo "$a, ";
                        $a=$a+2;
                }
        ?>
        </body>
        </html>
```

4. Write a PHP code to display the simple interest.

```
<html>
<head>
<title> Calculate simple Interest </title>
</head>
<body>
        <B> Calculate Simple Interest</B> <br/>
```

```
<form method="POST">
```

```
        Enter Principal:<input type="Number" name="p"/> <br/>
        Enter Time:<input type="Number" name="t"/> <br/>
        Enter Rate: <input type="Number" name="r"/> <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";
?>
</body>
<html>
```

Database Connectivity

Database: To organize the data in systematic and manageable form, the database is required. The database is the system where data and information are stored and organized systematically.

- The stored data can be retrieved easily when required.
- Data are stored in tabular form.
- The number of columns and rows forms the table.

The column of the table is known as a field that has a unique field name and the row of the table is known as records which represent individual information. Each table has a unique field known as a primary key. Each of the records of the primary field is different which make distinct from other records for identification.

MySQL is the most popular database system used with PHP. With PHP, we can connect to and manipulate MYSQL databases. It is a database system used on the web that runs on a server. It can be used for organized databases for both small and large applications. It is comparatively more fast, reliable, and easy to use than other databases. It uses standard SQL and compiles on a number of platforms.

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
```

```
echo "Connected successfully";
?>
```

Web References:

- https://www.javatpoint.com
- https://www.w3schools.com
- https://www.tutorialspoint.com
- https://www.google.com
- https://www.wikipedia.org

COMPUTER SCIENCE

Grade: XII

PROGRAMMING IN C

REFERENCE NOTE

Unit Wise NEB Important Questions for Computer Science XII

Unit-4- Programming in C - 8 Marks

1.) What is C? Write its Features.
2.) What are advantages and disadvantages of C programming Languages?
3.) What is a Data type? Explain the type of data with examples.
4.) What is variable? Explain its types.
5.) What is operator? Write its types and explain any four them.
6.) What is control structure? Write different between break and continue statement with examples.
7.) What is looping? Write different between while and Do while loop with examples.
8.) Define the term array. What is string? Explain any four string handling function with example.
9.) What is functions? Write its features and describe its types.
10.) What is recursion technique? Explain with one example.
11.) What is concept of storage? Differentiate between automatic storage and external storage.
12.) What is a Structure? Explain with one examples.
13.) Differentiate between array and structure.
14.) Differentiate between structure and union.
15.) Differentiate between array and pointer.
16.) Define the term call by value and call by reference with examples.
17.) What is pointer? Explain with examples.
18.) Differentiate between Structure and Pointer.
19.) Differentiate between sequential access and random access techniques of data file.
20.) Differentiate between fprintf and fscanf function.

IMP Selected C Programs

1. Write a program to input any three numbers and find out which one is largest numbers.
2. Write a program to display day using the switch statement depending upon the number entered. i.e. input 1 for Sunday, 7 for Saturday.
3. Write a program to input a number and check whether it is prime or not.
4. Write a program to display the sum of even and odd numbers from 1 to 50.
5. Write a program to accept the age of 10 different employees and count the number of employee.
    a. Whose age is more than or equal to 60
    b. Whose age is less than 35
6. Write a program to enter elements for 3x3 matrixes and Display its sum.
7. Write a program to accept 10 different numbers in array and sort in descending order.
8. Write a program to input a number and find out that number is palindrome or not.
9. Write a program to input the names of N numbers of students and sort them in alphabetical order.
10. Write a program to read a number and make the sum of individual digits and print using recursion technique.
11. Write a program that reads different names and address into the computer and rearrange the names into alphabetical order using the structure variables.
12. Write a program to read N students record store them in data file and display the content in appropriate format by using fprintf and fscanf function.

Unit 4- Programming in C

What is C? Explain its development process.

C is a high level language because no need for any architecture knowledge in normal English form.
C is a compiler because it can translate the whole program at a time so we can call compiler. C is structured programming language. It is called also procedural oriented programming language, function oriented language, module programming language. It is simple, reliable and easy to use.

What are the features of C? What are its advantages and disadvantages?

C is a computer language and a programming tool which has grown popular because programmers preferred it. It is a tricky language but a masterful one.
The C programming languages has the following ***features:***

i) It has small size.
ii) It has extensive use of function call.
iii) It is a strong structural language having powerful data definition methods.
iv) It has low level (Bit Wise) programming available.
v) It can handle low level activities.
vi) Pointer makes it very strong for memory manipulations.
vii) It has level constructors.
viii) It can produce efficient programs.
ix) It can be complied on variety of computers.

Advantage of C language

- It is machine independent programming language.
- It is easy to learn and implement C language.
- It can be implemented from mobile device to mainframe computers.
- It is the mother of all modern programming language.

Disadvantage of C Language

- There is no runtime checking.
- It has poor error detection systems.
- On large programs, it is hard to fix errors.
- It does not support modern programming methodologies oriented programming language.

What is preprocessor? Explain with its types.

The compiler of C has a preprocessor built into it. Lines that begin with # are called pre-processor directives. Each C program must start with proper header files (i.e.<stdio.h>) starting with '#', sign, 'include' and a header file name enclosed within triangle brackets.

What is header file? List the header files with some functions, where are they used?

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.

- Header files commonly contain forward declarations of classes, subroutines, variables, and other identifiers.
- The header file in c is called standard library functions.
- The entire header file has the extension .h
- A header file is used to define constants, variables, macros and functions that may be common to several applications.
- The updating and reading data of any function can be performed by using the header files.

Some of the frequent used header files are explain below:

S.N.	Header File	Description	Main Functions
1	stdio.h	Standard input and output	fpen(), fclose(), rename(), gets(), puts(), getchar(), scanf(), printf() etc.
2	conio.h	Old MS-DOS compiler header file, used for console input & output.	getch(), getche()
3	math.h	Mathematical calculation in C program.	sin(x), cos(x), log(x), pow(x,2), sqrt(x), cbrt(x), ceil(x), floor(x) etc.
4	complex.h	Complex arithmetic	cpow(x,p), csqrt(x), ctan(x), ctanh(x), cabs(x)
5	string.h	String / Words manipulation function	strlen(y), strcpy(z,y), strcmp(z,y), strcat(z,y), strupr(y), strlwr(y) etc.
6	ctype.h	Character manipulation type header file	toupper(y), tolower(y), isupr(y), isspace(), isalnu(y), toascii(y) etc.
7	stdlib.h	General purpose standard library.	rand(), malloc(), calloc(), abort(), exit() abs(), free() etc.

Fundamentals of C

1) What are the character set used in C?

A group of alphabetic, numeric and other characters that have some relationship with C programming language and recognized by compiler is called Character set. A character set can also contain additional characters with other code values.

The keywords, identifiers and other variables are constructed by using character set. The character set consists of following elements.

1) Alphabets:	= Uppercase (i.e. A,B,.....Y,Z) and Lowercase (i.e. a,b,. .....y,z)
2) Digits:	= 0,1,2,. 8,9
3) Special Symbols	= + - * / = ( ) , { } < > ' " ! # % & _ ~ \ ; : ?
4) White space characters:= blank, new line, tab etc.	

2. Define the term identifier, keywords and Tokens.

Identifiers:- Identifiers can be defined as the name of the variables, functions, arrays, structures etc created by the programmer. They are the fundamentals requirement of any language. The identifiers are defined according to the following rules:

- Identifiers consists letters and digits.
- First character must be an alphabet or underscore.

- Upper case and lowercase are allowed but not same, i.e. Text not same as text.
- Only one special character underscores (_) will used.
  For example, int a_b; Where ***a*** and ***_b*** are valid identifiers.

Keywords:-

Keywords are the reserved words which have standard, predefined meaning in C language. Keywords cannot be used as names for the variables or other user defined program elements. There are 32 keywords available in C. ***common examples are as follows.***

auto	double	if	static	break	else	int
struct	case	enum	long	switch	char	extern
const	float	near	typedef	register	union	continue
far	return	unsigned	default	for	short	void
do	goto	signed	while			

Tokens:

In a C source code, 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 components 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.
- Punctuation characters such as [,] , { , } , (,) , ; , : are also tokens.

2) Explain data types used in programming with examples.

Data types:

It is the set of keywords to declare variables. A set of data that specifies the possible range of values in a program and stored in memory are called data types. Data types are used to define variables before use it.

Types of data types in C.

1) Primary data types
2) Secondary data types

Primary Data Types: The basic fundamental of data having unit feature on C programming is called Primary Data Type. Example

Data Type	Type	Memory Require	Format Specifies
Char	Character	1 byte	% C
Int	Integer	4/2 byte	%d
Float	Floating point number	4 byte	%f
Long	Floating number	4 byte	%ld
Double	Large floating point number	8 byte	%lf
long double	Very large floating number	12 byte	%lf

Variable:

Variable are simply names that can change the value while executing a program. It allocates memory space inside a memory of computer. A variable can have only one value assigned to it in every time of execution of the program. Its value can change in different executions.

- ***Rules for variable declaration***
  - ✓ They must always begin with a letter, although some systems permit underscore as the first character.
  - ✓ White space is not allowed.
  - ✓ A variable should not be a keyword.

- ✓ It should not contain any special characters.

Types of variable

1. Numeric Variable: The variable that store numeric data only is called numeric variable. The numeric data may be whole number or fractional number. Examples are integer, floating point and double.

2. String Variable: The variable that stores character data only is called string variable. The string data may be single character or string. Examples are character, array of character (string), table of string.

Constant variable:

A constant is fixed entity. It does not change its value during the entire program execution. Constants can be classified as:

1. Integer constants
2. Floating point constants
3. Character constants
4. String constants
5. Symbolic Constants
6. Escape sequence constants

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. Most of the specifier used by printf and scanf functions. Types of mostly used specifier are explained below.

- ❖ **Escape Sequence:** They are a type of specifier. These non printable characters are used to format text on the output screen. These escape sequence character are place after backslash \.

Escape sequence	***Name***	***Meaning***
\'	Single quote	It prints ' in output
\"	Double quote	It prints " in output
\n	New line	It creates new line in output display
\t	Tab	It creates tab or 8 spaces in place of \t

- ❖ **Format Specifier:** The output and input data are display and receive in specific pattern. Format specifier uses the token % and character(s) after it. It is used to format for all types of data i.e. integer, float, character and string.

Format Specifier	***Used by scanf() function***
%d , % i	Signed integer + or – number o to 9
%f	Scans floating point numbers.
%s	String, Collection of character i.e. word
%c	Character, one single key stroke.

Operator:

An operator is a symbol that operates on a certain data type. The operator generally remains between the two operands. An expression is a combination of variables, constants, and operators written according to the syntax of the language. The data items that operators act upon are called operands.

Types of operator

1. Arithmetic Operator(Binary Operator)
2. Relational Operator (Comparison Operator)
3. Logical Operator (Boolean Operator)
4. Assignment Operator
5. Increment and Decrement Operators (Unary Operator)
6. Conditional Operator (Ternary Operator)
7. Bitwise Operator
8. Comma Operator
9. Size of Operator

1. Arithmetic Operator

The arithmetic operators perform arithmetic operations and can be classified into unary and binary arithmetic operations. The arithmetic operators can operate on any built-in data type. A list of arithmetic operators and their meanings are given below:

Operator	Meaning
+	additional or unary plus
-	subtraction or unary minus
*	multiplication
/	division
%	modulo division (returns remainder after division)

2. Relational Operator

The relational operators help to compare two similar quantities and depending on their relation, take some decisions. If the condition is true, it evaluates to an integer 1 and zero if the condition is false. The basic types of relational operator are:

Operator	Meaning
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to

3. Logical Operator

The logical operators are used to give logical value either true or false. They compare or evaluate logical and relational expressions. There are three logical operators.

Operator	***Meaning***	***Examples***
&&	Logical AND	(a>b) && (a>c)
||	Logical OR	(a>b) || (a>c)
!	Logical NOT	!(a==b)

4. Increment and Decrement Operators:

The increment and decrement operators are very commonly used in C language. The increment operators and decrement operators are extensively used in the loops using structures such as for, while, do, etc. The syntax of the operators is given below.

++<variable name>	Pre increment
--<variable name>	Pre decrement
<variable name>++	Post increment
<variable name)--	Post decrement

The pre increment operator increases the value of the variable by 1 and then the processing does whereas post increment first processes and increase the value of it by 1.

5. Conditional Operator:

A conditional operator is very rarely used. This can be carried out with the conditional operator (? :) An expression that makes use of the conditional operator is called a conditional expression. This can replace the if-else statement. The syntax of conditional operator is:

Expression_1 ? expression_2: expression_3

During evaluating the conditional expression, expression_1 is evaluated at the first step. If expression_1 is true (nonzero), then expression_2 is evaluated and this becomes the value of the conditional expression.

Library function:

The special functions that are well defined in C programming languages are called library functions such as printf(), scanf(),strlen(), sqrt(), tolower(), toupper(), getchar(), putchar() etc.

Control Structure:

Control structures are those programming constructs which control the flow of program statements execution in a program. Types of Control Structure

i) Branching / Decision (Selective Control Structure)
ii) Looping (Repetitive Control Structure)
iii) Jumping (Unconditional Control Structure)

1. Decision (Selective) Control Structure

It is mainly used for decision making. It is also called conditional statements. Selection is made on the basis of condition. We have options to go when the given condition is true or false. The flow of program statements execution is totally directed by the result obtained from checking condition.

Types

a) Conditional Statements

i. if statements:

It is used to execute an instruction or block of instructions only if a condition is fulfilled.

Syntax,

```
if(condition)
{
    Statements;
}
```

E.g. WAP to read a number and find even or odd by using if().

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,r;
    printf("enter the number");
    scanf("%d",&a);
    r=a%2;
    if(r==0)
    {
        printf("number is even %d",a);
    }
    if(r!=0)
    {
        printf("number is odd %d",a);
    }
    getch();
}
```

ii. <u>if else statements</u>

If the condition is true then the if() portion statements are evaluated otherwise else part of the statements are evaluated.

Syntax,

```
if( condition)
{
    Block of statements;
}
```

```
else
{
        Block of statements;
}
```

E.g. WAP input any two numbers and display the largest one.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int  a,b;
printf("enter the number");
scanf("%d",&a);
printf("enter the number");
scanf("%d",&b);
if(a>b)
{
        printf("A is greatest number",a);
}
else
{
        printf("B is greatest number",b);
}
getch();
}
```

iii. if() else if() statements

When we have two or more condition to be checked in a series we can use if else if statement. It is also known as multiple conditional statement or multipath conditional statement /if else ladder.

Syntax,

```
if(conditiona 1)
{
        Statements  1;
}
else if(condition  2)
{
        Statement 2;
}
else if(condition n-1)
{
        Statement n-1;
}
else
{
        Statement n;
}
}
```

e.g. WAP to find the largest number among three input number .

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a,b,c;
printf("\n Enter any three number");
scanf("%d",&a,&b,&c);
if(a>b&&a>c)
{
        printf("\n Largest number is %d",a);
}
else if(b>a&&b>c)
{
        printf("\n Largest number is %d", b);
}
else
{
        printf("\n Largest number is %d",c);
}
getch();
}
```

iv. Nested if else statements

An entire if else statement written within the body of if part or else part of another if else statement is called nested if else statement. It is used when a condition is to be checked inside another condition at a time in the same program to make decision.

Syntax,

```
if(condition 1)
{
        if(condition 2)
        {
                Statements 1;
        }
        else
        {
                Statements 2;
        }
}
else
{
        Statement 3;
}
```

E.g. WAP that reads marks of five subject and calculate total mark and percentage. Also awards the division on the basis of following criteria.

Percentage	***division***
p>=75	***distinction***
p>=60 and <75	***first***
p>=45 and <60	***second***
p>=35 and <45	***third***
otherwise	failed

```
#include<stdio.h>
#include<conio.h>
Void main()
        {
                int eng,nep,comp,acc,eco,total;
                Float per;
        printf("Enter the five subject mark");
        scanf("%d %d %d %d %d",&eng,&nep,&comp,&acc,&eco);
        total=eng+nep+comp+acc+eco;
        per=total/5;
        if(eng>=35&&nep>=35&&comp>=35&&acc>=35%%eco>=35)
        {
            if(per>=75)
              {
                printf("\n Distinction");
          else if(per>=60)
                {
                printf("\n First");
                }
           else if(per>=45)
                {
                printf("\n Second");
                }
           else
                {
                 printf("\n Third");
                }
        }
else
{
        printf("\n You are failed");
}
getch();
}
```

b) Switch case statements

The switch statement can be used instead of multiple if() else conditional statements. The switch control statement is mainly used to generate menu based programs.

Syntax,

```
Switch(expression 1)
{
        Case condition 1:
                Statements ….
                break;
                .
                .
        Case condition n-1:
                Statements…….
                break;
        default:
                statement n;
}
```

E.g. WAP which reads any two integer values from user and calculates sum, difference and product using switch case statement.

```
#include<stdio.h>
#include<conio.h>
Void main()
{
        int a,b,c,ch;
        printf("enter the two number");
        Scanf("%d",&a,&b);
printf("\n 1. Sum");
printf("\n 2. Difference");
printf("\n 3. Product");
        printf("\n Enter your choice");
        Scanf("%d",&ch);
switch(ch)
{
        case 1:
                c=a+b;
                Printf("\n Sum of two number is %d,c);
                break;
        case 2:
                c=a-b;
                printf("\n difference of two number is %d,c);
                break;
        case 3:
                c=a*b;
                printf("\n product of two number is %d,c);
                break;
        default:
                printf("\n Wrong choice");
```

```
        getch();
}
```

1. Looping Statement

The looping statement is also called repetitive or iterative control structure. Looping statements are the conditional control flow statements that repeats a certain portion of the program either a specified number of times or until a particular condition is satisfied or true.

Types of loop

i) For Loop ii) While Loop iii) Do while Loop

1**. For Loop:-**

The execution of for loop until the condition is true. The for loop is a entry control loop because it checks the condition at entry point.

Syntax,

```
for( initialization; condition; increment/ decrement)
        {
                // statements
        }
```

1. Write a program to print the natural number from 1 to 10.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i;
    for(i=1;i<=10;i++)
    {
            printf("/n%d",i);
    }
            getch();
}
```

2. Write a program to display even number from 1 to 20 and display their sum also.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, s=0;
    for(i=1;i<=20; i=i+2)
    {
            printf("/n even number are %d",i);
            s=s+i;
    }
    printf("sum of even number is %d", s);
    getch();
}
```

3. Write a program to find out sum of the cubes of first 10 numbers.

```
# include<stdio.h>
# include<conio.h>
```

```
void main()
{
    int i,c, sum=0;
    for(i=1;i<=10;i++)
    {
            c=i*i*i;
    sum=sum+c;
    }
printf("/n sum of cube is %d", sum);
    getch();
}
```

Nested for loop:

When for loop is declared inside another for loop is called nested for loop. The life of the inner for loop is depending over the outer for loop. If the outer for loop condition is true then inner for loop is evaluated. And will executes all the statements until condition is true if the inner for loop to be false then the outer for loop condition is reevaluated and so on.

For example

```
for( initialization; condition; increment/ decrement)
    {
        for( initialization; condition; increment/ decrement)
            {
                // statemetns
            }
    }
```

1. Display the following output.

10 20 30 40 50
10 20 30 40 50
10 20 30 40 50
10 20 30 40 50
10 20 30 40 50

2. WAP to display the following output.

55555
4444
333
22
1

Programs

```
# include<stdio.h>
# include<conio.h>
void main()
{
    int R,K;
    for (R=1;R<=5; R++)
    {
            for (K=10;K<=50;K=K+10)
            {
                    printf("\n %d",K);
            }
            printf("\n");

  }
    getch();
}
```

```
# include<stdio.h>
# include<conio.h>
void main()
{
        int R,K;
        for (R=5;R<=1; R++)
        {
                for (K=1;K<=R;K=K++)
                {
                printf("\n %d",R);
                }
                printf("\n");

         }
        getch();
}
```

While Loop:-

The while loop is also a entry control loop. While loop first checks whether the initial condition is true or false and finding it to be true, it will enter the loop and execute the statement.

Syntax,

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

1. Write a program to print even number from 1 to 100.

```
# include<stdio.h>
# include<conio.h>
void main()
{
        int i;
        i=2
        while<i<=100)
        {
                printf("%d\t", i)
                 i=i+2;
         }
        getch();
}
```

Do while loop:-

This loop is an exit control loop. This loop runs at least the once even though the termination condition is set to false. This loop test the condition at exit point hence it is called exit control loop.

The syntax of the loop is similar to while loop.

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

1. ***Write a program to display odd number from 100 to 1.***

```
#  include<stdio.h>
# include<conio.h>
void main()
{
        int i;
        i=99;
        do
        {
                printf("%d\t",i);
                i=i-2;

        }while(i>=0);
        getch();        }
```

2. ***Write a program to read the employee name, address for the N employee and display by using while loops.***

```
#  include<stdio.h>
# include<conio.h>
void main()
{
    int num;
    char ename[20],eadd[30];
    printf("\n enter the how many employee number");
    scanf("%d",& num);
    while(num>0)
    {
            printf("\n enter the name, address");
            scanf("%s%s",ename,eadd);
            num=num-1;
    printf("%s%s",ename,eadd);
    }
    getch();
 }
```

S.N.	***While loop***	***S.N.***	***Do while loop***
1	It is an entry controlled loop.	1	It is an exit controlled loop.
2	Testing starting in top	2	Testing started is bottom.
3	It has keyword while	3	It has keyword do and while.
4	If first condition true then the statement is executed otherwise no.	4	But in case at least one time executed statement if the condition is false.
5	Loop is not terminated with semicolon.	5	Loop is terminated with a semicolon.

6	Syntax
	while (expression)
	{
	//statements
	}
			Syntax
			do
			{
			//statements
			} while (expression);

The jump Statements

i) break statements
ii) continue statements
iii) goto statements

1) break statements:

The break statement is used to terminate loop or to exist form a switch. it can be used within a for, while, do while switch statement.

```
#  include<stdio.h>
# include<conio.h>
void main()
{
        int i;
        for(i=1;i<=10;i++)
        {
                if(i==5)
                {
                        break;
                }
        printf("%d",i);
        }
        getch();
}                                   OUTPUT:     1,2,3,4
```

The continue Statement

It skips the remaining statements in the body a while, for or do ..... while structure it proceeds with the next iteration of the loop. The continue statement is used to by pass the execution of the further statements.

```
#  include<stdio.h>
# include<conio.h>
void main()
{
        int num;
        for(num=1; num!=10; num++);
```

```
{
        if(num==7)
        {
                continue;
        }
        printf("%d",num);
        }
        getch();
        }
```

OUTPUT: 1,2,3,4,5,6,8,9

Go to statement:

The goto statement is used to send the pointer to the specified label. If the label is not defined then the goto statement will not work.

```
# include<stdio.h>
# include<conio.h>
void main()
{
  int num;
  Lab:
  printf("Enter a number");
 scanf("%d", &num);
 if(num<100)
  {
  printf("Mark can't be less then 100");
  goto Lab:
  }
else
{
  printf("Valid mark");
}
  getch();
}
```

Some important C programs

1. Reverse order
2. Factorial number
3. Fibonacci series
4. Prime or composite number
5. Even or odd number
6. Palindrome or not.
7. Sum of individual digits
8. Armstrong number or not
9. Multiplication table
10. Find list of prime number
11. Display perfect square number.

1. WAP to input any value and display the that value in reverse order.

```
#  include<stdio.h>
# include<conio.h>
void main()
{
        int i,r,n,x;
    printf("enter a number");
    scanf("%d",&n);
        x=0;
        while(n>0)
        {
            r=n%10;
            x=x*10+r;
            n=n/10;
        }
    printf("/n reverse number is%d",x);
        getch();
    }
```

2. WAP to input positive number and find its factorial number.

```
 # include<stdio.h>
  # include<conio.h>
  void main()
  {
        int i,n,f;
        printf("enter a number");
        scanf("%d", &n);
        if (n>0)
        {
            f=1;
            for(i=n;i>=1;i--)
            {
                f=f*i;
            }
            printf("/n The factorial value is %d",f);
            }
            else
            {
            printf("/n it is not a positive number");
            }
            getch();
        }
```

3. WAP to display the Fibonacci series. 1 1 2 3 5 8 13 ..... n.

```
#include<stdio.h>
#include<conio.h>
void main()
{
Int i,n;
 int x,y,z;
x=0,y=1,z=0;
printf("Enter the number");
scanf("%d",&n);
for(i=1;i<n;i++)
{
 printf("%d",z);
 z=x+y;
 x=y;
 y=z;
}
getch();
}
```

4.WAP read a number and to check the number is prime or not.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,num;
i=2;
printf("Enter the number");
scanf("%d",&num);
while(i<=num-1)
{
if(num%i==0)
{
        printf("Not a prime number");
        break;
}
i++;
}
printf("number is prime");
getch();
}
```

5.WAP to find out even number from 1 to 100 and find their sum also.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int i,sum=0;
        for(i=2; i<=100;i+2)
        {
                printf("\n Even number are %d",i);
                sum=sum+i;
        }
        printf("\n Sum of even numbers is %d",sum);
getch();
}
```

6.WAP to input a number and find out that number is palindrome or not.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a,n,r,x;
        x=0;
        printf("\n Enter the any number");
        scanf("%d",&n);
        a=n;
        while(n>0)
        {
                r=n%10;
                x=x*10+r;
                n=n/10;
        }
        if(a==x)
        {
        printf("Number is palindrome");
        }
        else
        {
        printf("Number is not palindrome");
        }
        getch();
}
```

7.WAP to input a positive number and find out the sum of its individual digits.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int i,r;n,x;
        printf("\n Enter a number");
        scanf("%d",&n);
        if(n>0)
        {
                x=0;
        while(n>0)
        {
                r=n%10;
                x=x+r;
                n=n/10;
        }
        printf("\n the sum of digits is %d",x);
        }
        else
        {
        printf("\n It is not a positive number");
        }
        getch();
}
```

8.WAP to input a number and check it is Armstrong number or not.

```
#include<stdio.h>
        #include<conio.h>
        void main()
        {
                int i,r,n,x,a;
                printf("Enter a number");
                scanf("%d",&n);
                a=n;
                if(n>0)
                {
                        x=0;
                        while(n>0)
                        {
                                r=n%10;
                                x=x+r*r*r;
                                n=n/10;
                        }
                        if(a==x)
                        {
```

```
                printf("\n Armstrong number is %d",a);
            }
            else
            {
                printf("\n Number is not Armstrong %d",a);
            }
        }
    else
    {
        printf("\n It is not positive number");
    }
        getch();
    }
```

9.WAP to display the multiplication table of given number.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int,i,n,m;
    printf("Enter the value ");
    scanf("%d",&n);
    for(i=1;i<=10;i++)
    {
    m=n*i;
        printf("\n %d * %d = %d",n,i,m);
    }
    getch();
}
```

10.WAP to display all prime numbers upto 1000.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,r,j,i;
    for(j=1;j<=1000;j++)
    {
        for(i=2;i<j;i++)
        {
            if(j%i==0)
            {
            break;
            }
        }
        if(i==j)
```

```
        {
                printf("\t%d",j);
        }
                getch();
}
```

11. WAP to display all perfect square numbers from 100 to 500.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int s,t;
        printf("The square numbers between 100 to 500 are");
        for(s=100;s<=500;s++)
        {
                for (t=1;t<=s;t++)
                {
                        if(s==t*t)
                        {
                                printf("%d  x %d =  %d \n",s,s,t);
                        }
                }
                getch();
        }
```

Arrays and String Function:

Arrays : An array is a collection of data of the similar type all of which are referred by a single variable name. for example, instead of using 50 individual variables to store name of 50 students, we can use an array to store name of 50 students.

Advantage of arrays.

- It is easier for handling similar types of data in a program.
- It is efficient for solving problems like sorting, searching, indexing etc.
- It is easier to solve matrix related problems.
- Graphics manipulations can be easily be done using array.

Disadvantages of arrays.

- It is not possible to hold dissimilar type of data in an array.
- It is difficult to visualize the multi dimensional array.
- It is static in nature so it is difficult to define the size of array during running time.

There are two types

1. **One/signal dimensional:** The values on an array variable assigned in one row and more than one column are called signal dimensional array.

   Syntax: type array_name[max. size];

   Example int n[10];

   int age[]= {18,12,19,20,16,16,17};

2. **Two/Double dimensional:** Two dimensional arrays are capable of storing data in multiple row and columns.

   Syntax: type array_name[No.Rows] [No.Cols];

   Example int n[10][5];

```
int matrix[3][3]= {{0,1,2},{3,4,5},{6,7,8}};
```

Program: ***Write a program to read 50 students marks and display them.***

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int i, M[50];
        printf("Enter the 50 student Marks"):
        for(i=1; i<=50; i++)
        {
                scanf("%d",&M[i]);
        }
        printf("The numbers are:");
        for(i=1; i<=50; i++)
        {
                printf("%d\t",M[i]);
        }
}
```

Program: Write a program to input 5 numbers with constant values initialization in array to display the sum.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int  i,sum=0;
        int num[5]={5,10,15,20,25};
        for(I=0;I<5;I++)
        {
                printf("%d\t",num[i]);
                sum=sum+num[I];
        }
        printf("\n Sum of 5 numbers is:%d",sum);
        getch();
}
```

Program: Write a program to input the age of 20 students and count the number of students having age in between 20 to 25.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int i, age[10],c=0;
        for(i=0;i<20;i++)
        {
                printf("enter age of students:");
                scanf("%d",&age[I]);
```

```
        }
        for(i=0;i<20;i++)
        {
                if(age[i]>=20&& age[i]<=25)
                {
                        c++;
                }
         }
                printf("Total number of students having age between 20 to 25 is %d",c);
                getch();
}
```

Program: Write a program to find the largest number among 'n' numbers.

```
#include<stdio.h>
#include<conio.h>
Void main()
{
        int i, n, num[100],max;
        printf("\n Enter the size of array not more than 100");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter array elements");
                scanf("%d",&num[i])
        }
        max=num[0];
        for(i=1;i<n;i++)
        {
                if(num[i]>max)
                {
                        max=num[i];
                }
        }
        printf("\n Largest number in array is %d",max);
        getch();
}
```

Program: ***Write a program to read a matrix, store it in array and display it.***

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        clrscr();
        int I,J, matrix[3][4];
        printf("Enter a matrix of 2x4:\n\n");
        for(I=1; I<=3; I++)
```

```
        {
                for(J=1; J<=4; J++)
                {
                        scanf("%d",&matrix[I][J]);
                }
        }
        Printf("\n The elements of matirx are: \n\n"):
        for(I=1; I<=3; I++)
        {
                for(J=1; J<=4; J++)
                {
                        printf("%d\t",matrix[I][J]);
                }
                printf("\n");
        }
}
```

Program: ***Write a program to calculate the average age of 10 students.***

```
#  include<stdio.h>
# include<conio.h>
void main( )
{
        int avg, i, sum=0;
        int age[10];
        for (i=1; i<=10; i++)
        {
                print("\n Enter age");
                scanf("%d',&age[i]);
        }
        for (i=1; i<=10; i++)
        {
                sum=sum+age[i];
                avg=sum/10;
        }
        printf("\n Average age=%d", avg);
}
```

Program: ***Write a program to accept the age of 10 different employees and count the no. of employee.***

i) Whose age is more than or equal to 60

ii) Whose age is less than 35

```
#  include<stdio.h>
# include<conio.h>
void main( )
{
        int age[10];
        int count1=0;
```

```
        int count2=0;
        int i;
        for (i=1; i<=10; i++)
        {
                printf("Enter the ages");
                scanf("%d", &age[i]);
        }
                for (i=1; i<=10; i++){
                        if (age[i]>=60)
                        {
                                count1++;
                        }
                        else
                        {
                                        if(age[i]<=35)
                                        {
                                        count2++;
                                        }
                         }
                    }
        printf("Employees above or equal to 60 %d",count1);
        printf("Employees under or equal 35 %d",count2);
}
```

Program: ***Write a program to store N numbers in array and print out the sum with the entire array variable.***

```
#  include<stdio.h>
# include<conio.h>
void main( )
{
        int j, sum=0;
        int n,num[10];
        clrscr();
        printf("Enter the how many no. you want to enter"):
        scanf("%d",&n);
        for (j=1; j<=n; j++)
        {
                printf("number stored %d",num[i]);
                sum=sum+num[j];
        }
        printf("\n Sum is %d",sum);
}
```

Program: ***Write a program to accept 10 different numbers in array and sort in descending order.***

```
#  include<stdio.h>
# include<conio.h>
void main( )
{
        clrscr( ):
        int array[10];
        int i, j ;
        for (i=1; i<=10; i++)
        {
        printf(" Enter the data serially);
        scanf("%d",&array[i]);
        }
        int temp;
        for( i=1; i<=10; i++)
        {
                for (j=i+1; j<=10; j++)
                {
                        if (array[i] < array[j])
                        {
                                Temp=array[i];
                                array[i]=array[j];
                                array[j]= temp;
                        }
                }
        }
        printf("Sorting data in descending order\n");
        for( i=1; i<=10; i++)
        {
                printf("%d\n",array[i]);
        }
}
```

Program: ***Write a program to store twelve numbers in double dimensional array and print out the values in table with row wise addition.***

```
#  include<stdio.h>
# include<conio.h>
void main( )
{
        int j, k;
        int num[3][4]={1,2,3,4,8,10,11,5,9,6,2,8}
        clrscr();
        printf("Data stored \t\t sum\n");
        for (j=1; j<=3; j++)
        {
                int sum=0;
```

```
                        for (k=1; k<=4; k++)
                        {
                                printf("%d \t",num[j][k]);
                                sum=sum+num[j]+[k];
                        }
                                printf("=%d',sum);
                                printf('\n");
                        }
                }
```

Program: WAP to enter elements for 2x2 matrix and display its transpose.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        int a[2][2];
        int i,j;
        printf("\n Enter number for 2x2 matrices");
        for(i=0;i<2;j++)
        {
        for(j=0;j<2;j++)
                {
                printf("Enter number");
                scanf("%d",&a[i][j]);
                }
        }
        printf("\n Transpose of Matrix is");
                for(i=0;i<2;j++)
                        {
                        for(j=0;j<2;j++)
                                {
                                printf("%d",a[i][j]);
                                }
                        printf("\n");
                        }
                getch();
        }
```

Program: WAP to enter elements for 3x3 matrix and display its sum.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        int m1[3][3], m2[3][3],
```

```
int i,j;
printf("\n Enter number for 2x2 matrices");
for(i=0;i<3;j++)
{
for(j=0;j<3;j++)
        {
        printf("Enter number for first matrix");
        scanf("%d",&m1[i][j]);
        }
}
for(i=0;i<3;j++)
{
for(j=0;j<3;j++)
        {
        printf("Enter number for Second matrix");
        scanf("%d",&m2[i][j]);
        }
}

printf("\n Sum of Two Matrix is");
        for(i=0;i<3;j++)
                {
        for(j=0;j<3;j++)
                {
                        printf("%d",m1[i][j]+ m2[i][j]);
                }
                printf("\n");
        }
getch();
}
```

String Function:

The strings are manipulated by specific string function. These are inbuilt functions and defined within string.h header file.

1. strlen( ): It returns the number of character present in the string.
   Syntax:strlen(string);

Program: ***Write a program to store string in array variable and find the length.***

```
#  include<stdio.h>
# include<conio.h>
# include<string.h>
void main( )
{
        int length;
        char n[ ]= "Everest";
        clrscr();
        length= strlen(n);
```

```
        printf("\n string=%s",n);
        printf("\n length=%d",length);
    }
```

2. strrev( ): It helps to reverse the character of the string.
   Syntax:strrev(string);
3. strupr( ): It converts lowercase letters in string to uppercase.
   Syntax:strupr(string);
4. strlwr( ): It converts uppercase letters in string to lowercase.
   Syntax:strlwr(string);
   strcpy( ): It is used to copy the content of one string to another.Syntax:strcpy(target,source);
5. strcat( ): It is used to concatenate source string to the target string.
   Syntax:strcat(target,source); or strcat(source,target);
6. strcmp( ): It compares two strings on following basis.
   Syntax:strcmp(string1,string2);

Program: ***Write a program to show use of trcpy,strrev,strupr and strlwr.***

```
#  include<stdio.h>
# include<conio.h>
# include<string.h>
void main( )
{
        char source[25];
        char target[25];
        printf("Enter the any word"):
        gets (source);
        strcpy(target,source);
        printf("Copied word is= %s \n",target);
        printf("Reverse word is=%s \n",strrev(source));
        printf("Word with capital letters= %s \n",strupr(source));
        printf("Word with small letters= %s \n",strlwr(source));
}
```

Program: ***Write a program to read two strings in array and concatenate string.***

```
#  include<stdio.h>
# include<conio.h>
# include<string.h>
void main( )
{
        clrscr( );
        char source[10]= "Rajesh";
        char target[10]= "Hamal";
        strcat (source,target);
        printf("\n %s", source);
}
```

Program: ***Write a program to read two strings in array and compare two strings and check that string is palindrome or not.***

```
#  include<stdio.h>
# include<conio.h>
# include<string.h>
void main( )
{clrscr( );int a;
        char string1[20];
        char string2[20];
        printf("Enter a word");
        scanf("%s",string1);
        strcpy(string2,string1);
        a=strcmp(string2, strrev(string2));
        if (a==0)
        {
                printf("palindrome");
        }
        else
        {
                printf("Not palindrome");
        }
}
```

Function:

Functions are the building block of statements, which takes some data, manipulate them and can return a value. Bug free function can be used repeatedly from other part of the program. There are two types of functions:

1. ***Library Functions (built in or ready made):*** C has the facilities to provide library function for performing some operation. These functions are present in the c library and they are predefined. for eg.
    a. scanf( );
    b. printf( );
    c. getchar( );
    d. putchar( );
2. ***User Defined functions (Defined by user according to need):*** user can creates their own function for performing this type of a function are called user define function.

Syntax of user defined functions

```
# include<stdio.h>
# include<conio.h>
void main( )
{
function ( );
.....
.....
}
```

```
function ( )
{
body of the function
}
```

WAP to calculate simple interest using function

```
#include
float interest(float); //function declaration
int main()
{  float si;
   si=interest(); //function call
   printf("Simple interst is %.2f\n",si);

}
float interest() //function definition
{
   float p,t,r,i;
   printf("Enter Principal, Time and Rate");
   scanf("%f%f%f",&p,&t,&r);
   i=(p*t*r)/100;
   return i; //function return value
}
```

WAP to calculate area of rectangle using function.

```
#include
int area (void);
void main()
{
int a;
        a = area();
printf("area is %d",a);

}
int area()
{
int l,b,ar;
        printf("Enter length and breadth");
        scanf("%d%d",&l,&b);
ar = l*b
        return ar;
}
```

Advantage:

1. Big programs can be divided into smaller module using functions.
2. Program development will be faster.
3. Program debugging will be easier and faster.
4. Use of functions reduce program complexity.
5. Program length can be reduced through code reusability.
6. Use of functions enhance program readability.
7. Several developer can work on a single project.
8. Functions are used to create own header file i.e mero.h
9. Functions can be independently tested.

With passing arguments

Program: Write a program to find out the area of circle through given radius as argument using function.

```
# include<stdio.h>
# include<conio.h>
        float area(float);
        void main( )
        {clrscr(); float rad,res;
        printf("enter the radius of a circle");
        scanf("%f",rad)
        res=area(rad)
        printf("your required area is %f ",res);
        }
        float area(float a)
        {
        float mm,pi=3.14;
        mm=pi*a*a;
        return(mm);
        }
```

Program: Write a program in C to create a function to pass two numbers as an argument and to return a sum to the calling function.

```
# include<stdio.h>
# include<conio.h>
        int add(int,int);
        void main( );
        {
                clrscr();
                int a,b,c;
                printf("enter any two numbers");
                scanf("%d%d",&a,&b);
                c=add(a,b);
                printf("The sum of a and b is= %d",c);
                }

                int add(int x, int y)
                {
                int sum;
                sum=x+y;
                return(sum);
                }
```

Without passing arguments

Program: Write a program to find out sum and square of two input number without passing arguments function.

```
# include<stdio.h>
# include<conio.h>
        int sum( );
        int square( );
        void main()
        {
          sum( );
         square( );
        }
        int sum( )
        {
                int a,b, sum;
                printf("enter any two number");
                scanf("%d%d",&a,&b);
                sum=a+b;
                printf("sum is=%d',sum);
                return(sum);
        }
         int square( )
        {
                int a,b, square;
                printf("enter any number");
                scanf("%d%d",&a);
                square=a*a;
                printf("square is=%d',square);
                return(square);
        }
```

Recursive Function:

The function which performs recursion is called recursive function. Recursion is a process by which a function calls itself repeatedly until some specified condition has been satisfied.

Those function which calls itself is known as recursive function and the concept of using recursive functions to repeat the execution of statements as per the requirement is known as recursion. The criteria for recursive functions are:

1. The function should call itself.
2. There should be terminating condition so that function calling will not be for infinite number of time.

Program: ***Write a program to calculate factorials by using recursion process.***

```
#include<stdio.h>
#include<conio.h>
int  fact(int);
void main( )
{
        clrscr();
        int
        printf("
        Enter a
        number"
        );
        scanf("
        %d",&n)
        ;
        printf("The factorial of %d is %d",n,fact(n));
}
int fact(int n)
{
        if(n<=1)
        return (1)
        else
         {
        return(n*fact(n-1));  or          z=(n*fact(n-1));
          }                               return z;
}
```

Program: ***Write a program read a number and make the sum of individual digits & print using recursion technique.***

```
#include<stdio.h>
#include<conio.h>
int sum(int)
void main()
{
        clrscr();
        int n;
        printf("Enter the any number");
        scanf("%d",&n);
        printf("The sum of individual digits is %d",sum);
int sum(int n)
{
        int a;
        if(n>0)
        {
```

```
            a=n%10;
            n=n/10;
            ruturn(a+sum(n));
            }
}
```

WAP to calculate sum of n-natural number using recursion/recursive function.

```
#include<stdio.h>
int sum (int);
void main()
{
   int n,s;
   printf("Enter any number");
   scanf("%d",&n);
   s = sum(n);
   printf("Sum is %d\n",s);

}
int sum (int n)
{
   if (n<=0)
      return 0;
   else
      return (n+sum(n-1));
}
```

Accessing a Function:

There are two types of accessing a function.

1. ***Call by value:*** Only the values of arguments are same to the function and any change made to the formal arguments do not change the actual arguments.

Program: *Write a C program try to exchange two values by using call by value accessing function.*

```
# include<stdio.h>
# include<conio.h>
int swap (int, int)
void main( )
{
        int a,b;
        a=10
        b=20
        printf("Value of a=%d and b=%d",a,b);
        swap (a,b);
        printf("Value of a=%d and b=%d",a,b);
}
int swap(int x, int y)
{
        int r;
        r=x;
        x=y;
        y=r;
}
```

2. ***Call by reference:*** When we pass address to a function the parameters receiving the address should be pointer. The process of calling a function by using pointer to pass the address of the variable is known as call by reference.

Program: *Write a C program to exchange two values by using call by reference accessing function.*

```
#  include<stdio.h>
# include<conio.h>
int swap(int *, int *)
void main( )
{
        int a,b;
        a=10
        b=20
        printf("Value of a=%d and b=%d",a,b);
        swap(&a,&b);
        printf("Value of a=%d and b=%d",a,b);
}

int swap(int *x, int *y)
{
        int r;
        r=*x;
        *x=*y;
        *y=r;
}
```

Structure and Union

Structure:

Structure is a collection of variables under a single name. As an example, if we want to store data with multiple data types like roll number, student name, address, phone number and class of a student then C supports structure which allows us to wrap one or more variables with different data types. Each variable in structure is called a structure member. To define a structure, we use struct keyword.

Syntax:

```
struct  struct_name
{
        Structure_members(s);
};
```

Structure Declaration

```
struct  student
{
        int  roll;
        char name[50];
        int age;
        char section;
        float  height;
};
```

We can also declare a structure

```
struct  student
{
        int  roll;
        char name[50];
        int age;
        char section;
        float  height;
} s1,s2;
```

We can also declare array of variables at a time of declaring a structure as:

```
struct  student
{
        int  roll;
        char name[50];
        int age;
        char section;
        float  height;
} s[100];
```

Union:

Unions like structure contain members whose individual data types may differ from one another. However the member that composes a union all share the same storage area within the computer memory were as each member within a structure is assigned its own unique storage area.

Syntax

```
Union  union_name
{
Union_member (s);
}
```

S.N.	Structure	S.N.	Union
1	Structure is designed by using 'struct' keyword.	1	Union is designed by using 'union' keyword.

2	*Syntax:* struct structure_name
	{
	Data_type member1;
	Data_type member1;
	}
		2	*Syntax:* union union_name
			{
			Data_type member1;
			Data_type member1;
			}

3	All the member of the structure variable can be processed at a given time.	3	Only one member of the union variable can be processed at a time because only one member of the union variable can be active at a time.
4	We use structure variable if memory is large and have to store values all of the variables.	4	We use union variables if memory is less and have to store one variable in one of the declared variables or members.
5	Structures are broadly used in programming.	5	Unions are not used broadly as much as structures.
6	Structure declaration takes large amount of spaces to store data and values.	6	Union declaration shares the same area of memory to save storage space.

Example 1, Write a C program to read different structure variables and display them.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        struct student
        {
                int roll;
                char name[30];
                char section;
                float height;
        } s1
        printf("\n Enter roll numer");
        scanf("%d",&s1.roll );
        printf("\n Enter name:");
        gets(s1.name);
        printf("\n Enter section:");
        scanf("%c",&s1.section);
        printf("\n Enter height");
        scanf("%f,&s1.height);
        printf("\n You have entered:\n);
        printf(" Roll Numer %d Name is %s section %c and height is %f",
        s1.roll,s1.name,s1.section,s.height);
}
```

Program: *Write a program to input the employee name and their basic salary of n employees and display the record in proper format.*

```
#include<stdio.h>
#include<conio.h>
void main ()
{
        struct employee
        {
                char name[10];
                int salary;
        };
struct employee e[100],t;
int i,j,n;
printf("\n How many records");
scanf("%d", &n);
for(i=0; i<n; i++)
{
        printf("\n Enter name of employee");
        gets(e[i].name);
```

```
            printf("\n Enter salary of employee:");
            scanf("%d",&e[i].salary);
        }
        printf("\n Employee \t \t salary);
        printf("\n ============================");
        for(i=0; i<n; i++)
        {
            printf("\n %s \t\t\t %d",e[i].name,e[i].salary);
        }
        printf("\n============================");
    }
```

Program: *Write a program that reads different names and addresses into the computer and rearrange the names into alphabetical order using the structure variables.*

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main ()
{
int i,j,n;
    struct student
    {
        char name[50];
        char address[50];
    };
    struct student s[200],t;
    printf("\n how many records do you want store?");
    scanf("%d",&n);
    for (i=0; i<n; i++)
    {
            printf("\n Enter name of the student:");
            gets(s[i].name);
            printf("\n Enter address of the student:");
            gets(s[i].address);
    {
    for (i=0; i<n; i++)
    {
        for (j=0; j<n-1; j++)
        {
            If(strcmpi (s[j].name,s[j+1].name)>0)
            {
                t=s[j];
                s[j]=s[j+1];
                s[j+1]=t;
```

```
                        }
                    }
                }
                Printf("\n\n The records of the student after sorting\n");
                Printf("\n Name of student          Address       ");
                for (i=0; i<=n; i++)
                {
                    printf("\n %s %s", s[i].name,s[i].address);
                }
            }
```

Pointer

A pointer is a variable that a points to a references a memory location where data is stored. Each memory cell in the computer has an address which can be used to access its location. A pointer variable points to a memory location rather than a value.

- We can access and change the contents of the memory location.
- A pointer variable contains the memory location of another variable.
- The asterisk tells the compiler that you are creating a pointer variable.

The pointer declaration syntax is as shown below.

```
Pointer_type *pointer_variable_name;
```

For e.g. `int *p;`

Address (&) and indirection (*) operator

The address (&) operator and immediately preceding variable returns the address of the variable associated with it. Hence, the address of (&) operator returns the address of memory location in which the variable is stored.

The indirection (*) operator is used to extract value from the memory location stored in the particular memory location whose address is stored in pointer variable.

The syntax of declaring address operator to pointer variable is as follows.

Pointer_variable = &variable_name;

For Example

```
int *ptr, num=25;
ptr     = &num;
```

Program: Write a complete program to display address and value of the pointer variable.

```
#include<stdio.h>
void main( )
{
    int *p;
    int age=17;
    p=&age;
    printf("\n Value of age is %d", age);          output= 17
    printf("\n Value of age is %d",*p);          output=17
    printf("\n Value of age is %d",*(&age));     output=17
    printf("\n Address of age is %u",p);          output=2686784
    printf("\n Address of age is %u",&age);      output=2686784
}
```

Pointer to Arithmetic

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int *p;
        int age= 17
        p=&age;
        printf(" \nValue of age is %d",age);                    output=17
        printf("\n Increment on age is %d",++age);              output=18
        printf("\n Address of age is %u",p);                    output=2686784
        printf("\n Increment in pointer is %u",++p);            output=
}
```

Write a C program to increment pointer.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int n[ ]={10,20,30,40};
        int *aptr=n;
        int i;
        printf("\n Pointer notation");
        for (i=1; i<=4; i++);
        {
                printf("*(aptr+%d)=%d\n",i*(aptr+i));          output  =10
        }                                                               =20
}                                                                       =30
                                                                        =40
```

Assignment pointer value to function

Program: Write a program to pass pointer variable to function sum them and display after returning it.

```
        #include<stdio.h>
        #include<conio.h>
        Int add(int *p, int *k)
        void main( )
        {
                int a,b,c=0;
                printf("\n Enter two numbers");
                scanf("%d %d",&a,&b);
                c=add(&a,&b);
                printf("\n Sum of two numbers %d and %d is %d",a,b,c);
        }
```

```
add (int  *x, int *y)
        {
                int r;
                r=*x+*y;
                return (r);
        }
```

Array of pointers:

Program: Write a C program to assign array, place these array in pointer variable and display array value along with its address.

```
#include<stdio.h>
void main( )
{
        int i, arr[10];
        Int *ptr[10];
        for (i=1; i<=10; i++)
        {
                Arr[i]=i;
        }
        for (i=1; i<=10; i<=10; i++)
        {
                ptr[i]=&arr[i];
        }
        for (i=0; i<=10; i++)
        {
                printf("\n value arr[%d] stored at address of %u",arr[i],ptr[i]);
        }
}
```

Pointer to Pointer

Program: Write a program to demonstrate the use of pointer to pointers:

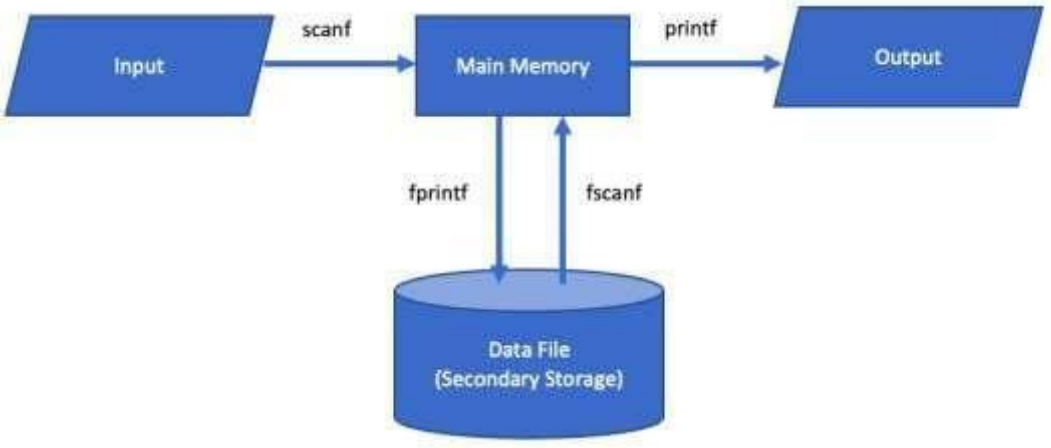
```
#include<stdio.h>
Void main( )
{
        int a=10, *b, **c;
        b=&a;
        c=&b;
        printf("\n Value of a is %d",a);                output= 10
        printf("\n Value of a is %d",*b);               output= 10
        printf("\n Value of a is %d",**c);              output= 10
        printf("\n address of a is %u",b);              output= 2686788
        printf("\n address of a is %u",&a);             output= 2686788
        printf("\n address of b is %u",c);              output= 2686784
        printf("\n address of b is%u", &b);             output= 2686784
}
```

Advantage of Pointer

- Runtime memory creation.
- Runtime memory deletion.
- Hard Access through pointer.
- Data Structure Based Pointer.
- Applications of Pointer Ms-word, Excel, Access, SQL.

Working with files

Data file much application required that information be written is stored on the memory device in the form of a data file. Thus, data file access and alter that information whenever information in C.



Sequential and Random Access File

Sequential 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.

Random Access File:

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 function fseek( ), lseek( ), rewind( ) and ftell( ) are the some of the common examples of random access files.

Opening, Reading, Writing and Appending on/from Data File:

Once the file pointer has been declared, the next step is to open file. There is an inbuilt function to open a file. The function fopen( ) is used to create a steam for use and links of new open file. This function return a file pointer and takes two parameter one for name of file and other for mode for the file. The syntax is as follows;

```
FILE    *f;
f = fopen ("file_name.extension",  "mode_of_open");
```

The modes of the data file are as follows:

S.N.	*Mode*	*Description*
1	"r" / "rt"	It opens a text file to read only mode.
2	"w" / "wt"	It creates a text file to write only mode.
3	"a" / "at"	It appends text to already data containing file.
4	"r+t"	It opens a text file for read and write mode.
5	"w+t"	It creates a text file for read and write mode.
6	"a+t"	It opens or creates a text file and read mode.
7	"rb"	It opens a binary file for read only mode.
8	"wb"	It creates a binary file for write only mode.
9	"ab"	It opens or create a binary file for append mode.
10	"r+b"	It opens a binary file for read and write mode.
11	"w+b"	It creates a binary file for read and write mode.
12	"a+b"	It opens or creates a binary file for append mode.

Functions

1. fputc= Store character into the file.
2. fputs= Store string in to the file.
3. fgetc= fetch character from the file.
4. fgets= fetch string from the file.
5. fwrite= store data (structure) in to the file.
6. fread= fetch data (structure) from the file.
7. fprintf= store data in to file.
8. fscanf= fetch variable from the file.

Opening a data file

Syntax:
FILE *fptr
fptr = fopen (“filename” , ”mode”)
Where, File name can be “library.txt”, “student.dat” ..etc
Mode:
“w” to write/store data in a data file.
“r” to display/read/retrieve/access data from a datafile.
“a” to add/append data in existing datafile.

Store/write data

Syntax:
fprintf(fptr , ”format specifiers” ,variables);
Eg; suppose if we want to store name, disease, age and bed number of a patient then, it is written as

```
fprintf(fptr , "%s %s %d %d", n, d, a, b);
```

Where, variable are initialized as:

```
char n[10], d[10];
int a, b;
```

Program example

1) Create a datafile "patient.txt" and store name, disease, age and bed number of a patient.

```
#include<stdio.h>
void main()
{
   char n[10], d[10];
   int a, b;
   FILE *fptr;
   fptr = fopen("patient.txt","w");
   printf("Enter name, disease, age and bed number");
   scanf("%s %s %d %d", n, d, &a, &b);
   fprintf(fptr,"%s %s %d %d\n", n, d, a, b);
   fclose(fptr);
}
```

[Note: This program will store only single record to store multiple record we have to use loop as following programs.

2) Create a datafile "student.txt" and store name, class and marks obtained in 3 different subject for few students/n-students.

```
#include<stdio.h>
void main()
{
   char n[10];
   int c, e, ne, m, i, num;
   FILE *fptr;
   fptr = fopen("student.txt","w");
   printf("How many record?");
   scanf("%d",&num);
   for(i=1;i<=num;i++)
   {
   printf("Enter name class and 3 marks");
   scanf("%s %d %d %d %d",n, &c, &e, &ne, &m);
   fprintf(fptr,"%s %d %d %d %d \n",n, c, e, ne, m);
   }
   fclose(fptr);
}
```

3) Create a datafile "student.txt" and store name, class and marks obtained in 3 different subject until user press "y" / as per user requirement.

```
#include<stdio.h>
int main()
{
   char n[10],ch[3];
   int c, e, ne, m;
   FILE *fptr;
   fptr = fopen("student.txt","w");
```

```
do
{
printf("Enter name class and 3 marks");
scanf("%s %d %d %d %d",n, &c, &e, &ne, &m);
fprintf(fptr,"%s %d %d %d %d\n",n, c, e, ne, m);
printf("Press Y to continue");
scanf("%s",ch);
} while (strcmp(ch,"Y") == 0 || strcmp(ch,"y")==0);
fclose(fptr);
}
```

Add/Append data

1) A datafile "student.txt" contain name, class and marks obtained in 3 different subject of few students. Write a C program to add 200 more records.

```
#include<stdio.h>
 void main()
{
   char n[10];
   int c, e, ne, m, i;
   FILE *fptr;
   fptr = fopen("student.txt","a");
   for(i=1;i<=200;i++)
   {
   printf("Enter name class and 3 marks");
   scanf("%s %d %d %d %d", n, &c, &e, &ne, &m);
   fprintf(fptr,"%s %d %d %d %d \n",n, c, e, ne, m);
   }
   fclose(fptr);
 }
```

2) A datafile "student.txt" contain name, class and marks obtained in 3 different subject of few students. Write a C program to add more records until user press "y" / as per user requirement.

```
#include<stdio.h>
  void main()
{
   char n[10], ch[3];
   int c, e, ne, m;
   FILE *fptr;
   fptr = fopen("student.txt","a");
   do
   {
   printf("Enter name class and 3 marks");
   scanf("%s %d %d %d %d", n, &c, &e, &ne, &m);
   fprintf(fptr,"%s %d %d %d %d\n", n, c, e, ne, m);
   printf("Press Y to continue");
```

```
scanf("%s",ch);
```

```
} while (strcmp(ch,"Y") == 0 || strcmp(ch,"y")==0);
fclose(fptr);
}
```

Read/Display/retrieve/access data from a datafile

Syntax:

fscanf(fptr , "format specifiers" ,variables);

Eg; suppose if we want to display/read name, disease, age and bed number of a patient from data file then, it is written as

```
fscanf(fptr , "%s %s %d %d", n, d, &a, &b);
```

Where, variable are initialized as:

```
char n[10], d[10];
int a,b;
```

EOF: End of file

1) A d

```
#include <stdio.h>
void main()
{
   char  n[10];
   int c, e, ne, m;
   FILE *fptr;
   fptr = fopen("student.txt","r");
   printf("Name\tPercentage\n");
   while(fscanf(fptr,"%s %d %d %d %d",n,&c,&e,&ne,&m) != EOF)
   {
      printf("%s %d  %d  %d  %d", n, c, e, ne, m);
   }
   fclose(fptr);
}
```

2) A datafile "student.txt" contain name, class and marks obtained in 3 different subject of few students. Write a C program to read and display only records whose name is Ram.

```
#include<stdio.h>
void main()
{
   char  n[10];
   int c, e, ne, m;
   FILE *fptr;
   fptr = fopen("student.txt","r");
   while(fscanf(fptr,"%s %d %d %d %d",n,&c,&e,&ne,&m) != EOF)
   {
     strlwr(n);
     if (strcmp(n,"ram") == 0)
     {
      printf("%s %d  %d  %d  %d", n, c, e, ne, m);
```

```
        }
      }
      fclose(fptr);
    }
```

3) A datafile "student.txt" contain name, class and marks obtained in 3 different subject of few students. Write a C program to read and display only records who pass in all subjects.

```
#include<stdio.h>
void main()
{
   char  n[10];
   int c, e, ne, m;
   FILE *fptr;
   fptr = fopen("student.txt","r");
   while(fscanf(fptr,"%s %d %d %d %d",n,&c,&e,&ne,&m) != EOF)
   {
     if (e>=40 && ne>=40 && m>=40)
      {
        printf("%s %d  %d  %d  %d", n, c, e, ne, m);
      }
   }
   fclose(fptr);
 }
```

4) A datafile "student.txt" contain name, class and marks obtained in 3 different subject of few students. Write a C program to read and display only records who fail in any one subject.

```
#include<stdio.h>
void main()
{
    char  n[10];
    int c, e, ne, m;
    FILE *fptr;
    fptr = fopen("student.txt","r");
    while(fscanf(fptr, "%s %d %d %d %d", n, &c, &e, &ne, &m) != EOF)
    {
      if (e<40 || ne<40 || m<40)
     {
       printf("%s %d  %d  %d  %d", n, c, e, ne, m);
     }
    }
    fclose(fptr);
}
```

5) A datafile "student.txt" contain name, class and marks obtained in 3 different subject of few students. Write a C program to read and display only name and percentage of all students

```
#include<stdio.h>
int main()
{
   char  n[10];
   int c, e, ne, m;
   float p;
   FILE *fptr;
   fptr = fopen("student.txt","r");
   while(fscanf(fptr, "%s %d %d %d %d", n, &c, &e, &ne, &m) != EOF)
   {
      p = (e+ne+m)/3;
      printf("%s %f", n, p);
   }
   fclose(fptr);
 }
```

6) A datafile "student.txt" contain name, class and marks obtained in 3 different subject of few students. Write a C program to read and display only records of all students who secure distinction.

```
#include<stdio.h>
int main()
{
   char  n[10];
   int c, e, ne, m;
   float p;
   FILE *fptr;
   fptr = fopen("student.txt","r");
   while(fscanf(fptr, "%s %d %d %d %d", n, &c, &e, &ne, &m) != EOF)
   {
      p = (e+ne+m)/3;
      if (p>=80)
      {
       printf("%s %d  %d  %d  %d", n, c, e, ne, m);
      }
   }
   fclose(fptr);
 }
```

Program: ***Write a program to read a line and store it in a data file and display the contents.***

```
#include<stdio.h.
#include<conio.h>
void main( )
char I;
FILE  *f,   *q;
```

```
        f = fopen("store.txt","w");
        while((i=getchar())!='\n')
        {
                fputc(i, f);
        }
        fclose( f);
        q=fopen ("store.txt","r");
        while((i=fgetc(q))!=EOF)
        {
        Printf("%c",i);
        }
}
```

Program: ***Write a C program to read N students record store them in data file and display the content in appropriate format by using fprintf and fscanf function.***

```
#include<stdio.h>
void main( )
{
  FILE  *f;
  f= fopen (''ratna.txt","w");
  int  i, rn,  n;
  char name[25],    add[35];
  printf("\n how many records?");
  scanf("%d", &n);
  printf("Enter %d student roll number, name and address:\n",n);
  for (i=o; i<n; i++)
  {
          scanf("%d %s  %s",&rn,name,add);
          fprintf( f, "%d\t%s\t%s\n ",rn,name,add);
  }
  fclose(f);
  printf("\n The %d data records",n);
  f= fopen ("ratna.txt","r");
  while (fscanf( f, "%d  %s  %s",&rn,name,add)!=EOF)
  {
          printf("\n %d\t%s\t%s",rn,name,add);
  }
  fclose(f);
}
```

Program: Write a program using C language that reads successive records from the new data file and display each record on the screen in an appropriate format.

```
#include<stdio.h>
#include<conio.h>
void main( )
```

```
{
        struct student.
        {
                char name[50];
                char add[80];
        };
        struct student    s;
        char next='Y' ;
        FILE  *fp;
        fp = fopen ("C:\\student.txt","w");
        while(next=='Y' || next=='y')
        {
                printf("\n Enter the name of the student");
                gets(s.name);
                printf("\n Enter the address of the student");
                gets(s.add);
                fwrite(&s, sizeof (s), 1, fp);
                printf("\n Do you want to write next record (Y/N)");
                                }
                                fclose(fp);
                                fp=fopen ("C:\\student.txt", "r");
                                printf("\n Name of student Address");
                                while (fread (&s, sizeof(s), 1,f(p)==1)
                                {
                                        printf("\n %s \t\t\t %s",s.name,s.add);
                                }
                                fclose (fp);
```

Program: ***Write a C program to read sentence until enter key pressed. Put every words in data file by using fputs( ) fucnction and display the content of file by using fgets( ) function.***

```
#include<stdio.h>
#include<conio.h>
Void main( )
{
        Char a[20];
        FILE  *f;
        f= fopen ("senten.doc", "w");
        printf("\n Enter sentence");
        gets(a)
        while (strlen (gets(a))>0)
        {
                fputs(a,f);
        }
```

```
                fclose(f);
                f = fopen ("senten.doc", "r");
                if (f== NULL)
                printf("\n Cannot open the file");
                else
                {
                        While (fgets (a,19,f)!=NULL)
                        {
                                printf("%s",a);
                        }
                }
                fclose(f);
        }
```

Program: ***Write a C program to read ages of N students, store them in age.txt file and find out the average age after reading from data file.***

```
#include<stdio.h>
#include<conio.h>
Void main( )
{
        int ag,n, i, sum=0;
        float avg;
        FILE *f;
        f= fopen("age.txt", "w");
        printf("\n How many student:");
        scanf("%d",&n);
        for(i=0; i<n; i++)
        {
                printf("\n Enter %d student ages",i+1);
                scanf("%d",&ag);
                putw(ag,f);
        }
        fclose (f);
        f= fopen ("age.txt", "r");
        printf("\n Calculating average age from file");
        for (i=0; i<n; i++)
        {
                sum=sum+getw(f);
        }
        avg = (float) sum / n ;
        printf("\n Average age of %d student is %f", n, avg);
        fclose(f);
}
```

Program: ***Write C program to read already created data file student.txt and display it contents if it opens otherwise print message "The file ……does not exists".***

```
#include<stdio.h>
#include<conio.h.
Void main( )
{
        char i;
        FILE  *f;
        f = fopen ("student.txt", "r");
        If (f== NULL)
        {
                printf("\n The student.txt file does not exists");
        }
        else
        {
                while ( ( i=fgetc(f))!=EOF)
                {
                        printf("%c",i);
                }
        }
        fclose(f);
```

Some IMP Notes

S.N.	Fwrite	S.N.	fprintf
1	Binary format writing	**1**	No binary format.
2	Size of data.	**2**	Size of data is not fix
3	Number of records.	**3**	No number of records.
4	We store the data in block form so not in format form.	**4**	Format string used to store the data in the format form.
5	Fixed block size.	**5**	No block size.

S.N.	Structure	S.N.	Pointer
1	Structure is use to store the difference type of data member like as char, int, float etc.	1	Where is pointer is use to store the address of another location (variable).
2	Structure is use to maintain the records.	2	Whereas pointer is use to create link list.
3	Structure can't be use to access the hardware.	3	Pointer can be use to access the hardware.
4	Structure can't be use to create memory run time.	4	But pointer are use to create the memory runtime.
5	Structure can be declared as pointer variable.	5	Pointer can be declared inside the structure.

S.N.	Break Keywords (Statement)	S.N.	Continue Keywords (Statement)
1	The break statement is use to terminate the loop unconditionally.	1	The continue statement is used return the pointer at the beginning of the loop.
2	The break statement can also be use inside the switch statement.	2	But the continue can be use inside the switch statement.
3	The break statement is use to terminate the loop.	3	The continue statement is use to repeat set of statements.

4	Example
	`void main( )`
	`{`
	`int i;`
	`for (i=1; i<=10; i++)`
	`{`
	`If (i==5)`
	`{`
	`break;`
	`}`
	`printf("%d",i);`
	`}`
	`}`
	Output= 1,2,3,4.
		4	Example
			`void main( )`
			`{`
			`int num;`
			`for (num=1; num!=10; num+1)`
			`{`
			`If(num==7)`
			`{`
			`continue;`
			`}`
			`printf("%d",num);`
			`}`
			`}`
			Output= 1,2,3,4,5,6,8,9.

S.N.	Array	S.N.	Pointer
1	Array is create continue memory location with same type.	1	Pointer creates only one block of memory but access all memory of another variable.
2	Array takes huge of memory.	2	Pointer takes less and less memory.
3	Array is to be fixed at the time of declaration.	3	Whereas pointer size can be change at the run time.
4	In array we can free the memory run time.	4	But using pointer the dynamic allocated memory can be make free at run time.
5	If we allocated huge sized of array and we get few information then unnecessary memory will be occupied.	5	But this problem can resolved by using the pointer.
6	Using array technique to accessing the array elements is more different comparison to pointer.	6	It is easier than array.

Web References:

3. https://www.javatpoint.com
4. https://www.w3schools.com
5. https://www.tutorialspoint.com
6. https://www.google.com
7. https://www.wikipedia.org

COMPUTER SCIENCE

Grade: XII

OBJECT ORIENTED PROGRAMMING (OOP)

REFERENCE NOTE

NEB Important Questions for Computer Science XII

Unit-5- Object Oriented Programming (OOP) - 6 Marks

1) ***What is OOP? Describe its features.***
2) ***What are the advantages and disadvantages of using OOP?***
3) Explain the term polymorphism and inheritance.
4) ***Difference between OOP and Structured Programming Language.***
5) Explain the Class, Object and Encapsulation.

UNIT 5. Object Oriented Programming

1. What are the programming methods or approaches of program development?

There are two approaches of program development. They are Procedure Oriented Programming and Object-Oriented Programming. Procedure Oriented Programming is a conventional method of programming and the Object-Oriented Programming is a modern or latest programming method.

2. What is Procedure Oriented Programming?

It is a conventional or old method of programming, in which the program is written into many small parts and combined together. In this approach, the functions are created and the data is not very crucial. Variables are created for the data handling and they are treated as the global and local variables. Creation of the variables inside of the sub programs is known as local variable and the creation of the variables in the main module is called the global variable. Global variables can be accessed from any modules but the local variables can be accessed only within the local modules. The alteration of data is very high.

3. What are the features of Procedure Oriented Programming?

The characteristics or features are as follow:

a) A large program is broken down into small programs or procedures.
b) It focuses on the functions rather than the data.
c) Variables are created as local and global.
d) The possibility of data alteration is very high, which is the main disadvantage of this approach.
e) It follows top down method.

4. What is Object Oriented Programming?

It is a modern approach of programming. It is highly known as OOP in short form. In this method, all the real world entities are treated as the objects and objects are collected in a class. Even the classes are controlled by the Super class. And by the inheritance feature, the changes on the super class are easily passed to its sub classes. Similarly, it was developed to overcome procedure oriented programming method and the data is given high priority rather than the functions. Data can be hidden, so that the possibility of data alteration is very less.

5. What are the characteristics of OOP?

a) Emphasis is given to the data.
b) Program are divided into multiple objects.
c) Functions and data are tied together in a single unit.
d) Data can be hidden to prevent accidental alteration.
e) It follows the bottom up approach.

6. What are the differences between Procedure Oriented and Object Oriented Programming?

Procedure Oriented	Object Oriented
1. Emphasis is given to procedures.	1. Emphasis is given to data.
2. Programs are divided into multiple modules.	2. Programs are divided into multiple objects.
3. It follows top-down method.	3. It follows bottom-up method.
4. Generally data cannot be hidden.	4. Data can be hidden.
5. It does not model the real world perfectly.	5. It models the real world perfectly.
6. Maintenance is difficult.	6. Maintenance is easy.
7. Code reusability is difficult.	7. Code reusability is easy.
8. Examples: FORTRAN, COBOL, Pascal, C, etc.	8. Examples: C++, JAVA, Smalltalk, etc.

7. Write short notes on the following:

a) Object

All the entities of a program used in OOP method are called objects. Here entities represent a group of people, teachers, students, books, cars, etc. Each entity or object does have an attribute called characteristics and the behavior or functions. For example, a car can be an object. The colour like blue, black, size, weight, etc. are the attributes or the characteristics, which distinguishes to it with other objects and move, turn, etc. can be the functions.

b) Class

Class is a user defined data type in OOP, which defines the data types for all the objects, which run under it. Or it collects the objects of its similar data types. For example, a class vehicle can have the objects like car, bus, truck, etc. Similarly a class school can have students, teachers, staff, etc.

c) Abstraction

It is a feature of hiding internal detail of any object. It provides only the interface to the user, which makes them easy to use but does not show the details of that object, how that works and how that is made. Due to this feature, OOP has become very secure platform for its data from being accidental alteration.

d) Encapsulation

It is a process of combining the data and functions together. OOP gives more emphasis on the data rather than the functions or procedures. Many functions can use the same data but the instruction given to the function to use any particular data and combining them together is the encapsulation. Due to its unrelated functions cannot use unnecessary data in the program.

e) Inheritance

Inheritance is the process of creating new classes based on the existing class. The new classes require the features of the main class called the Super class and it is provided through the feature called Inheritance. By the Inheritance feature Super class can coordinate with it's sub classes. It models the real world. It allows the extension and reuse of existing code without having to rewrite for the new created classes.

f) Polymorphism

It is a feature of OOP, which refers to the way of operating the same operator in different ways and different method or purpose. Operator overloading and the operation overloading are the examples or Polymorphism. For example '+' operator can be used for arithmetic operation and string concatenation both. This facility or feature is an example of Polymorphism. It reduces the number or keywords or operators.

8. What are the advantages and disadvantages of OOP?

Advantages:

a) Code repetition is reduced by the various techniques like inheritance.
b) Data is more secure due to the data hiding feature called abstraction.
c) Existing classes can serve as library class for further enhancements.
d) Division of a program into multiple objects makes the software development easier.
e) Software complexity is less.
f) Upgrading and maintenance of software is easy.
g) It perfectly models the real world system.
h) Code re-usability is much easier than the conventional programming system.

Disadvantages

a) Compiler and runtime overhead is high.
b) Software developer should analyze the problem in object oriented way.
c) Requires the mastery in software engineering and programming methodology.
d) Useful only for the large and complex projects.

Application of OOP

1. Expert System
2. Artificial intelligence
3. Management information systems.
4. Decision support system.
5. Computer based training and education
6. Object- oriented database.
7. Computer games.
8. Mobile applications.
9. Internet based applications.
10. Designing user interface for software.
11. Security System.

Web References:

- https://www.javatpoint.com
- https://www.w3schools.com
- https://www.tutorialspoint.com
- https://www.google.com
- https://www.wikipedia.org

COMPUTER SCIENCE

Grade: XII

SOFTWARE PROCESS MODEL

REFERENCE NOTE

NEB Important Questions for Computer Science XII

Unit-6- Software Process Model **- 6 Marks**

1. What is SPM? Explain the major activities of Software Development Process in brief.
2. ***What is Software Development Model? Explain the types of SDM with advantages and disadvantage.***
3. ***Who is the system analyst? Explain the role and characteristic of a system analyst.***
4. **Explain in detail about waterfall, prototyping and spiral model of software *development*.**
5. What is prototyping? Write down its advantages and disadvantages.
6. ***Define SDLC. Explain different phases of SDLC in brief.***
7. ***What do you mean by feasibility study? Why is it necessary before designing a system?***
8. Define system testing. Different between black box testing and white box testing.
9. ***What is algorithm and flowchart? Write difference between system and program flowchart.***
10. Differentiate between System analyst and System engineer.
11. Define the following terms. A) ***DFD*** B) ***ER diagram*** C) Case diagram D) UML E) System Flowchart

    F) ***Documentation*** G) Algorithm H) Pseudo code I) Decision table and decision tree

    J) Prototyping model K) Spiral Model

Unit 1- Software Process Model (SPM)

Software Project Concept

Program:

The program is a sequence of instructions. It is the set or collection of instructions.

Instruction:

An instruction is a command given to the computer to perform a certain specified operation on given data.

Software:

A set of programs written for a computer to perform a particular task is called software or the logical components or set of procedures to routines or instructions are called software is the interface between the computer and the user. It is a group of commands that tells the computer what to do and how to do it.

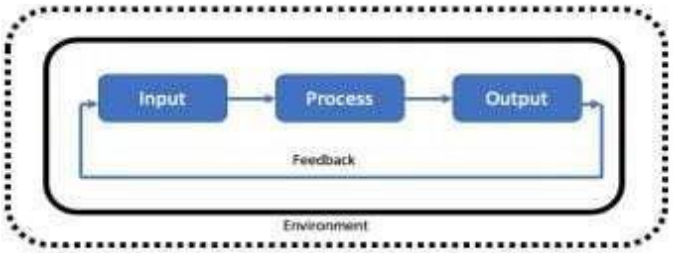
Project:

A project is a well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery).

Software Project Concept:

A software project is the complete procedure of software development from requirement identification to testing and maintenance, carried out according to the execution methodologies, in a specified period of time and budget in order to achieve intended software product.

Software Development Process



- Software development process defines a sequence of tasks that must be carried out to build new software.
- It groups the development activities into a sequence of phases.
- A phase in sequence can only commence on the previous phase has been completed.
- A report is produced at the end of each phase, describing what has been achieved and outlining the plan for the next phase.

Fundamental activities for the software development process are

1. **Software Specification:** The functionality or software and constraints on its operations must defined.
2. **Software design and implementation:** The software to meet the specifications must produce.
3. re must be validated to ensure that it does what the customerwants.
4. **Software evolution:** The software must evolve to meet changing customer needs.

System Development Life Cycle (SDLC)

System:

System is a set of interacting or interdependent components forming an integrated whole. A system can be described as a set of objects joined together for a common objective.

Development:

It is the process of step by step changing or growing of any program and system.

Information System:

Information system is a system which processes supplied/collected data and generates information that can be used for decision making at different levels.

SDLC

SDLC (Software/System Development Life Cycle) is an organized way to develop a software/system. System Development Phase or System Development Life Cycle or Software Development Life Cycle (SDLC) is a methodology used to develop, maintain, and replace software/information systems.

- *It is a systematic process of developing any software. It helps in establishing a system, or software or project, or plan. It gives an overall list of processes and sub-processes required for developing a system.*
- SDLC consists of a set of development activities that have a prescribed order. It is the development of software in chronological order.

System Development Life Cycle (SDLC), which is also known as Application Development Life Cycle, is a term used in system that describes the process of planning, creating, testing and deploying an information system.

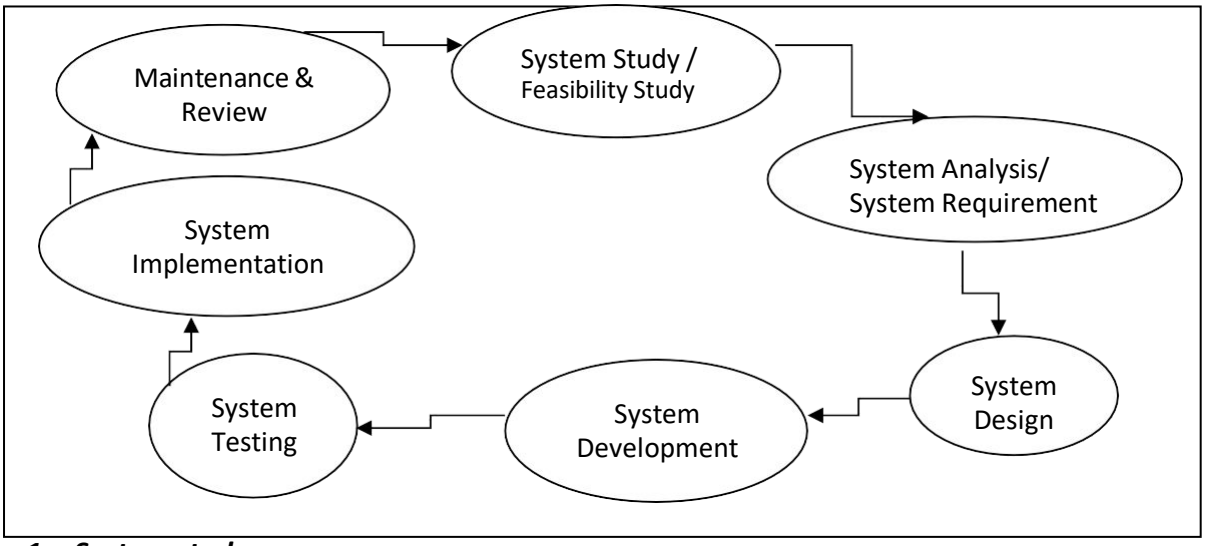
Importance and the necessity of SDLC

1. It helps to determine the needs of the user.
2. It supports constant communication between the developer and the user.
3. SDLC helps for easy identification of missing requirements.
4. It ensures that the software meets the needs of its users.
5. It supports proper analysis and design of the software.
6. It ensures proper development and testing.
7. Proper documentation support for future upgrade and maintenance.
8. It provide flexibility for adding features even after the software is developed.

SDLC Phases

The different **phases of SDLC** are as follows:

1. System Study or Preliminary Investigation and Feasibility study.
2. System analysis or Determination of system requirements.
3. System design.
4. System development or development of software.
5. System Testing.
6. System Implementation.
7. System Maintenance and Reviews or Evaluation.



1. System study:

A system is intended to meet the needs of an organization. Thus the first step in the design is to specify these needs or requirements. The top manager of the organization takes the basic decision to use a computer based (information) system for managing the organization.

During this phase, the development team focuses on completing three tasks:

- Survey the system by collecting the inputs from various sources.
- Analyzing the current system (manual or automated) in depth and developing possible solutions to the problem.
- Selecting to the best solution and defining its function with a feasibility study.

2. System analysis:

System analysis is the dissection of a system into its component pieces to study how those component pieces interact and work.

- System analysis is a term that collectively describes the early phases of development.
- It is defined as those phases and activities that focus on the business problem, independent of technology.

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 analyzes the information and proposes the following specifications.

i. Goals and objectives of the proposed system.
ii. Fundamental actions that must take place in the software.
iii. Outputs to be produced.
iv. Inputs to be used.
v. Processes to be performed.
vi. Interfaces to be provided.

vii. Organizational and other constraints' to be met.
viii. Performanceequements to be met.

Feasibility study:

Feasibility study is the most important activity in the system analysis phase. It analyses the proposed system from different aspects so that it makes us clear that how practical or beneficial the system will be to the organization. So it tells us whether the system is feasible to design nor not.

Need of feasibility study

- It determines whether the system meets the goal of the clients or not.
- It determines the strength and limitations before starting to develop the system.
- It focuses on the boundary of the system's outline.
- It suggests new opportunities through the investigations process.
- It provide quality information for decision making.
- To provide documentation of the investigated system.

The different levels of feasibility study are as:

1) ***Economic feasibility:*** it concerns with cost effectiveness of the system. The main objective of economic feasibility is to calculate approximate cost-both the development cost and the operational cost and the benefits from the system.
2) ***Technical feasibility:*** it concerns with the availability of the hardware, software and the support equipment for the complete development of the system.
3) ***Operational feasibility:*** it concerns with smooth operation of the system. It is all about the problems that may occur during operation of the system after its development.
4) ***Behavior feasibility:*** it concerns with behavior of the users and the society towards the new system. Generally, most of the traditional employees are not easily ready to upgrade them with the new system.
5) ***Social Feasibility:*** It is a determination of whether a proposed system will be acceptable to the people or not.
6) ***Management Feasibility:*** It is a determination of whether a proposed system will be acceptable to management or not.
7) ***Schedule(Time) feasibility:*** it is the process of splitting project into tasks and estimate time and resources required to complete each task. It determines the deadline to complete a system and schedule the task accordingly.
8) ***Legal feasibility:*** it concerns with legal issue of the system. If the system is illegal then the system designing is meaningless. Everything is measured whether it is legal or illegal. It considers copyright law, foreign law, foreign trade, tax, etc.

System design:

The next step is to develop the logical design of the system. During this phase, the logic of the system, namely, the information requirement of users, and use this to find the necessary database.

System design is concerned with the design of new system. It involves designing of various things such as output design, input design, files design, processing and general program design etc.

Logical Design: Theoretically designing of the system is called logical design. The system could be designed on the basis of the requirements.

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. It uses

algorithms, flowcharts, pseudo codes, decision table, decision tree, E-R diagram, Data flow diagram etc.

Theoretically designing of the system is called **logical design**. The system could be designed on the basis of the requirements.
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.

To create the logical design different kinds of tools are used.

- Algorithm
- Flowchart
- Pseudo codes
- Structured English
- Decision Table
- Decision Tree
- Data flow diagram
- E-R diagram

3. ***System development:*** after designing a logical diagram of a system then next step is to convert into program. This process is called system development. Flowchart, algorithm, Pseudo code, etc. are the outlines the procedures for taking the input data and processing it into usable output.

4. ***System testing:*** it is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. System testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.
5. ***Implementation:*** implementation involves testing the installed system, converting from the old system to the new one and training the users. This phase consists of implementation of the system into a production environment, and resolution of the problem identified in testing phase.
6. ***Maintenance and review:*** it begins after the system is implemented. Like any system, there is an ageing process that requires periodic maintenance of hardware and software. The content of the review will include objectives met, cost, performance, standards and recommendation.

System Analyst:
System analyst is person who is involved in analyzing, designing, implementing and evaluating computer-based information systems to support the decision making activities and operations of an organization.
A good system analyst is:

1. Understanding and commitment to the organization
2. People skills
3. Conceptual skills and
4. Technical skills

- *A system analyst is information specialist. To be a system analyst, one must be knowledgeable about the technical aspects of analyzing, designing and implementing computer-based systems.*
- *A system analyst is a person who conducts a study, identifies activities and objectives and determines a procedure to achieve the objectives.*

Designing and implementing systems to suit organizational needs are the functions of the systems analyst. One plays a major role in seeing the business benefits from computer technology.
An analyst is a person with unique skills. One uses these skills to coordinate the efforts of different types of persons in an organization to achieve business goals.
The characteristics (attributes) of system analyst are as follows:

1. Knowledge of organization.
2. Technical Knowledge.
3. Interpersonal Communication Skill.
4. Character and Ethics.
5. Problem-Solving Skill.
    1. Defining the problem
    2. Analyzing the problem

3. Considering Many alternatives

4. Evaluating many alternatives
5. Choosing the best alternatives

6. System analysis and Design skills

The roles of system analyst area as follows:

1. Change event
2. Investigator and event
3. Architect
4. Psychologist
5. Motivator
6. Intermediary and diplomat

Duties and Responsibilities of System Analyst

1. Defining Requirements
2. Prioritizing Requirements
3. Analysis and Evaluation
4. Solving Problems
5. Drawing up functional specification
6. Designing System
7. Evaluating System

SDLC (System Development Life Cycle):

Describes in Details:

1. System study or Preliminary Investigation :

In this stage, the development team studies the present and identifies the drawbacks. They interact with the users and gathers information from different sources to recognize the problems of present system.

2. System Analysis:

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 analyzes the information and proposes the following specifications.

i. Goals and objectives of the proposed system.
ii. Fundamental actions that must take place in the software.
iii. Outputs to be produced.
iv. Inputs to be used.
v. Processes to be performed.
vi. Interfaces to be provided.
vii. Performance requirements to be met.
viii. Organizational and other constraints' to be met.

3. Feasibility study:

Feasibility study is the most important activity in the system analysis phase. It analyses the proposed system from different aspects so that it makes us clear that how practical or beneficial the system will be to the organization. So it tells us whether the system is feasible to design nor not. Thus it is necessary before system design.

The different levels of feasibility study are as:

1. ***Economic feasibility:*** it concerns with cost effectiveness of the system. The main objective of economic feasibility is to calculate approximate cost-both the development cost and the operational cost and the benefits from the system.
2. ***Technical feasibility:*** it concerns with the availability of the hardware, software and the support equipment for the complete development of the system.
3. ***Operational feasibility:*** it concerns with smooth operation of the system. It is all about the problems that may occur during operation of the system after its development.
4. ***Behavior feasibility:*** it concerns with behavior of the users and the society towards the new system. Generally, most of the traditional employees are not easily ready to upgrade them with the new system.
5. ***Schedule feasibility:*** it is the process of splitting project into tasks and estimate time and resources required to complete each task. It determines the deadline to complete a system and schedule the task accordingly.
6. ***Legal feasibility:*** it concerns with legal issue of the system. If the system is illegal then the system designing is meaningless. Everything is measured whether it is legal or illegal. It considers copyright law, foreign law, foreign trade, tax, etc.

4. System Design:

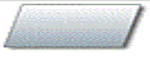
System design is concerned with the design of new system. It involves designing of various things such as output design, input design, files design, processing and general program design etc. This state consists of logical design and physical design of the system.

7. ***Logical Design:*** Theoretically designing of the system is called logical design. The system could be designed on the basis of the requirements.
8. ***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. It uses algorithms, flowcharts, pseudo codes, decision table, decision tree, E-R diagram, Data flow diagram etc.

System Design Tools:

The tools which are used to design the system in known as system design tools. They are used during system analysis and design phase of the system development.

1. **Algorithm:** An algorithm is defined as the finite sequences of instructions for solving a problem.
2. **Flowchart:** A flowchart is the pictorial representation of an algorithm which is classified into two types' system flowchart and program flowchart. The different symbols used in system flowchart are defined below:

Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

I) System flowchart:

System flowchart describes the internal architecture of a system that describes how data are moved inside the internal components of a system.

II) Program flowchart:

Program flowchart describes to solve the application types of real world problem.

3. DFD (Data flow diagram):

DFD is the logical diagram to describe the flow of data inside the components of system. It is easier to understand or grasp when being explained and most important to all, it is much more precise and less ambiguous than a narrative one. The main components are: process, data store, data flow, external entities.

4. Context Diagram:

It is combination of many other DFD. It is the highest level of DFD. It contains only one process, representing the entire system, the process is given the symbol circle. The external entities are denoted by rectangle. The flow of data is described by arrow.

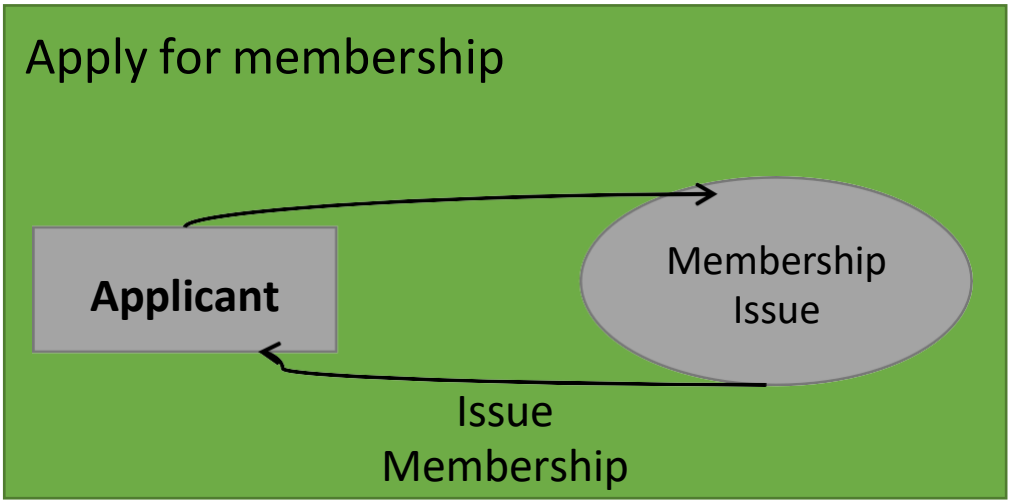


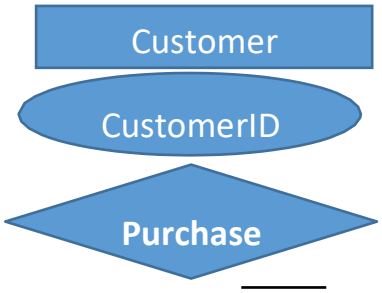
Fig.: The Context diagram of library membership

5. ER (Entity Relationship) diagram:

The E-R diagram is an overall logical structure of a database that can be expressed graphically. It was developed to facilitated database design and the simplicity and pictorial clarity of this diagramming technique have done great help in the designing part of database. The main components are attributes, entities and relationship.

The diagrammatic representation of entities attributes and their relationship is described by E-R diagram.

- **The main components of ERD are**
- Entity:
- Attributes:
- Relationship:
- Lines:



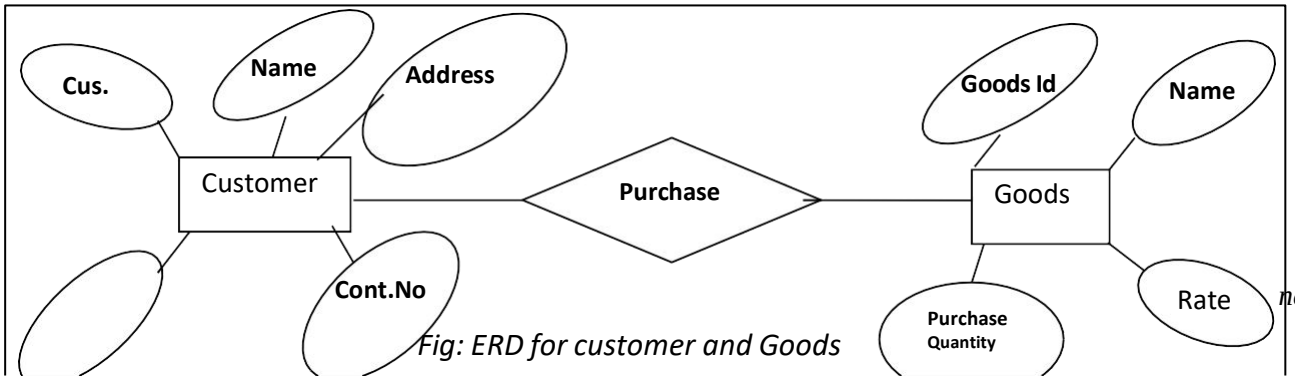
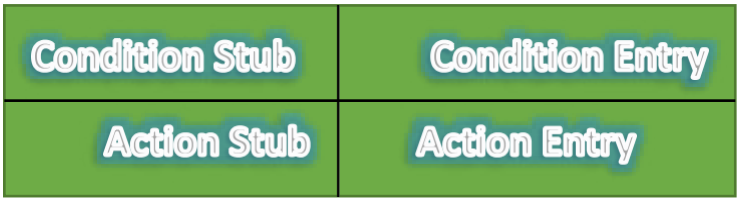


Fig: ERD for customer and Goods

6. **Case diagram**
   Computer aided software engineering tool is automatic computer based program that helps for software engineering and SDLC process. It is very fast and effective tools for the development of big scale software. It helps in analysis, design, implementation, testing and maintenance.
7. **UML**
   Unified Modeling Language is a standardized general purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the object management group. UML includes a set of graphic notation techniques to create visual models of object-oriented software.
8. **Decision Table:**
   A table allows us to identify the exact course of actions for given conditions in tabular form. Decision table is a tabular representation of the logic of a decision, which specifies the possible conditions for the decision and the resulting actions.
   Parts of Decision Table.



9. **Decision Tree:**
   Decision tree is also a technique to represent condition and actions in a diagrammatic form in computer. A decision tree allows us to identify the exact course of actions for given conditions in tree structures.
10. **Pseudo Code:**
    It is a kind algorithm for solving a problem and the instructions of pseudo code are written by using English phrase and mathematical expressions.

5. System Development:

Programmers begin to develop the program by using a suitable High Level Language. In System developments following processes are done.

- **i.** Convert logical structure in to programs in programming language.
- **ii.** Database is created.
- **iii.** User operational documents are written.
- **iv.** Users are trained.
- **v.** The internal documentation of a system is prepared.

6. System testing:

It is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. System testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

1. ***White box testing**:* white box testing of software is predicted on close examination of procedural details. Logical path through the software and collaborations between components are tested by providing test case that exercises specific sets of conditions or loops. It is used when the tester has access it the internal data structures and algorithms including the code that implement these.
2. ***Black box testing**:* black box testing treats the software as a black box without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary

value analysis, specification based testing, etc. it is also called functional testing because it tests whether a system is functioning or not.

7. Implementation:

Implementation involves testing the installed system, converting from the old system to the new one and training the users. This phase consists of implementation of the system into a production environment, and resolution of the problem identified in testing phase.

Types of Implementation:

1. ***Direct Conversion:*** All users stop using old system at the same time and then begin using the new. This option is very fast, less costly but more risky.
2. ***Parallel conversion:*** Users continue to use old system while an increasing amount of data is processed through the new system. Both the systems operate at the same time until the new system works smoothly. This option is costly but safe approach.
3. ***Phased conversion:*** Users start using the new system component by component. This option works only for systems that can be compartmentalized. This option is safe and conservative approach.
4. ***Pilot conversion:*** Personnel in a single pilot-site use the new system, and then the entire organization makes the switch. Although this option may take more time, it is very useful in big organizations where a large number of people make the conversion.

8. Maintenance and review:

It begins after the system is implemented. Like any system, there is an ageing process that requires periodic maintenance of hardware and software. The content of the review will include objectives met, cost, performance, standards and recommendation.

Types of Maintenance

1. ***Corrective Maintenance:*** it corrects the run time errors during the operation.
2. ***Adaptive Maintenance:*** It modifies or adds new features in the system.
3. ***Perfective Maintenance:*** It makes the system perfect, up-to-date and improve the life of the system.

System Development Model:

During software development or system development for organizations, a common process framework is established, defining a small number of framework activities that are applicable to all software projects, regardless of their size complexity. For a better paradigm of a software process, several models are designed and implemented. It is the choice of system analyst which model is used to achieve the goal. The different models are:

1. **Waterfall model:**

   Waterfall model is a systematic and sequential model to develop software that begins with requirements analysis to operation and maintenance. It describes a development method that is liner and sequential. It is an oldest type of model for software engineering. The fundamental processes of waterfall model are as follows:

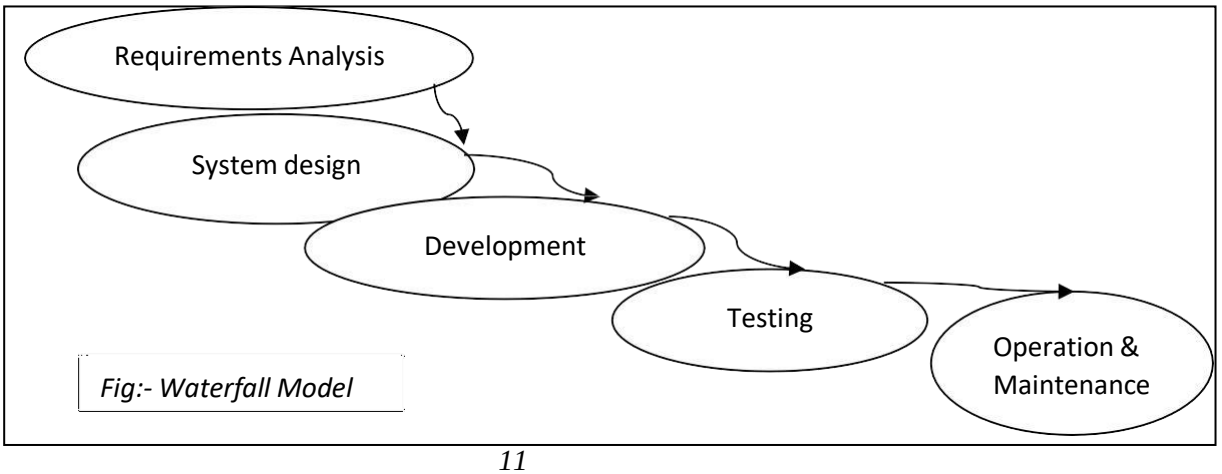


Fig:- Waterfall Model

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 partition the requirements to either hardware or software systems.
3. ***System Development:*** During this stage, the system design is converting into development.
4. ***Integration and system Testing:*** The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met.
5. ***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

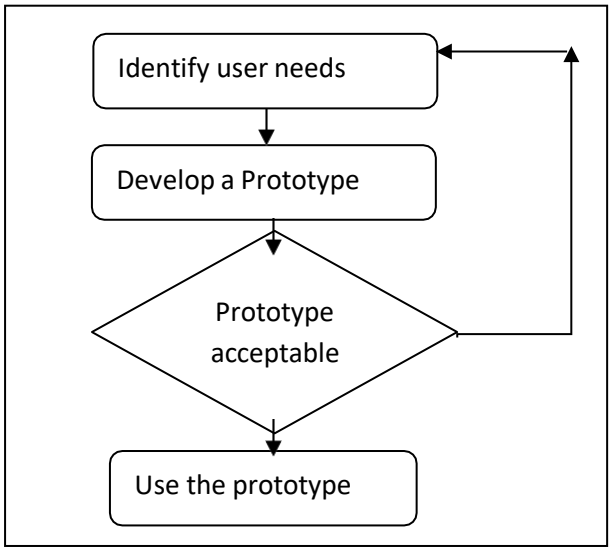
1. It is simple model suitable for small size project.
2. It is less expensive.

Disadvantages

1. It has no back track mechanism.
2. It is not suitable for large size project.
3. It has lack of proper documentation.

2. Prototyping model:

It is the iterative process of system development which is more appropriate for developing new system where there is no clear idea of requirements, inputs and outputs. These systems are then continuously modified until the user is satisfied.



1. ***Identify the user needs:*** the system analyst interviews the user to obtain an idea of what is required from the system.
2. ***Develop a prototype:*** the system analyst, working uses one or more prototyping tools to develop a prototype.
3. ***Determine if prototype is acceptable:*** the analyst educates the user in prototype use and provides an opportunity from becoming familiar with the system.
4. ***Use the prototype:*** the prototype becomes the operational system.

Advantages

1. The users get a better understanding of the system being developed.
2. Errors can be detected much earlier as the system is made side by side.
3. Quicker user feedback is an available leading to better solutions of the system.

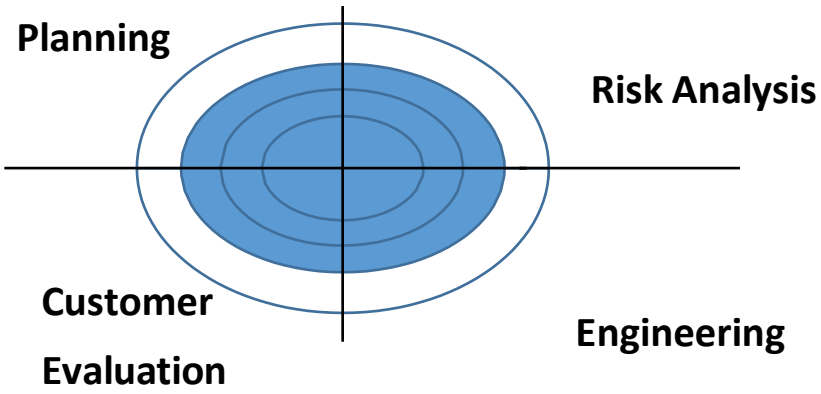
Disadvantages

1. It leads to implementing and repairing way of building systems.
2. It may increase the complexity of the system as scope of the system may expand beyond original plans.

3. Spiral system:

In this model, process is represented as a spiral rather than as a sequence of activities with backtracking. It is a software development process combining the elements of both waterfall and prototyping model. The spiral model is intended for large, expensive and complicated projects.

This is the most realistic model because it uses multidimensional approach for software development. The activities in SDLC are organized in a spiral structure that has many cycles which starts from the center of the spiral and goes out as it program and becomes matured. Each of the complete spiral segment is divided into four different attributes knows as



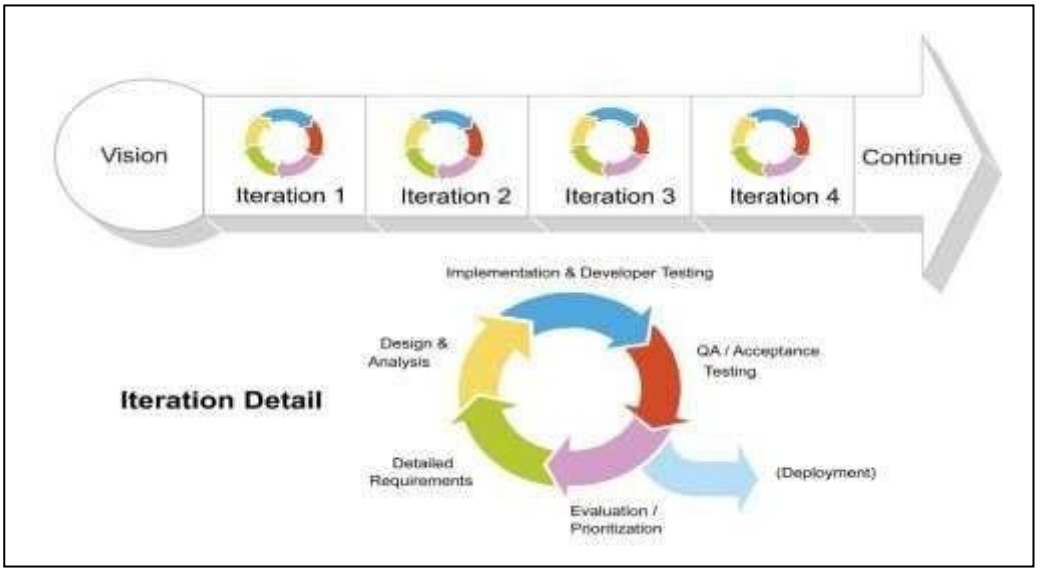
1. **Planning:** the project is reviewed and a decision made whether to continue with a further loop of the spiral. If it is decided to continue, plans are drawn up for the next phase of the project.
2. **Risk analysis:** for each of the identified project risks, a detailed analysis is carried out. Steps are taken to reduce the risk. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.
3. **Software development (Engineering):** after risk evaluation, a development model for the system is chosen.
4. **User evaluation:** specific objectives for the phase of the project are defined by the evaluation of users. Constraints on the process and the product are identified. And a detailed management plan is drawn up. ***Advantages***
   - It emphasizes quality.
   - It is effective for regular updating the system.
   - It emphasizes risk reduction techniques.

Disadvantages

- Full scale risk analysis requires training, skill so it may appropriate only for large projects.
- This model is relatively untested.

4. Agile Software Development

It is a software development method based on iterative and incremental development in which requirement and solutions evolve though collaboration between self-organizing, cross functional teams.



Documentation:

Documentation is the process of collecting, organizing, storing and maintaining a complete record of system and other documents used or prepared during the different phases of the life cycle of the system. It consists of the detail description about software requirements specification, feasibility report, and software designing report, description about input-output and processing mechanism, source code, comments, manuals, guides and effective help desk.

Types of Documentation

1. ***Internal Documentation:*** It is used by the system analyst and the programmer during development process. It is very useful for the development period and also useful in future for the modification and maintenance for the software.
2. ***External Documentation:*** It is used by the user during the running time of the software. It includes the detail description in terms of manuals, guide and help files. It is mainly deals with how to effective use of software.

Web References:

- https://www.javatpoint.com
- https://www.w3schools.com
- https://www.tutorialspoint.com
- https://www.google.com
- https://www.wikipedia.org

COMPUTER SCIENCE

Grade: XII

RECENT TRENDS IN TECHNOLOGY

REFERENCE NOTE

NEB Important Questions for Computer Science XII

Unit 7 Recent Trends in Technology

1. What is e-commerce? Explain its role to enhance the digital economy.
2. Write the impact of e-commerce technology in our society.
3. ***What is cloud computing and robotics?***
4. ***Define term e-learning and virtual reality.***
5. ***What is AI? Write its applications.***
6. ***What is big data? Write its advantage and disadvantages.***
7. ***What is e-governance? List out the objectives.***
8. Write Short notes on the following.
   a) IOT
   b) Mobile Computing
   c) Social Media
   d) e-medicine
   e) AI

Artificial Intelligence (AI)

AI stands for Artificial Intelligence. It is a concept of giving human-like intelligence to the machines. Though the computers do their work faster and better than the human beings, the intelligence of them is zero because they just follow the set of instructions given by the user. In case of wrong instruction, they do wrong processing. It is because they do not have intelligence of their own. So, the scientists are in research of giving them artificial intelligence, so that they can understand the natural languages of the human beings and interact. They can express their feelings and many more.

Components of AI

Different disciplines contributed their ideas, viewpoint, and techniques to plan the foundation of Al that acts as components of Al. Some of the major contributions of various disciplines an given below

1. **Philosophy:** It introduces the concept of logic and methods of reasoning and studying the mind as a physical system. It creates the foundation for learning language, and rationality. It also expresses knowledge-based action to be embedded into the machine to act with AI
2. **Mathematics:** It introduces the concepts of the formal representation of facts and proof, algorithms, computation, and reasoning with uncertain information.
3. **Economics:** It introduces the concepts of the formal theory of rational decision.
4. **Neuroscience:** It introduces the concepts of mental activity which can be introduced into the machine.
5. **Psychology:** It introduces the concepts of the brain as an information processing device and phenomenon of perception and sensory-motor control.
6. **Linguistics:** It introduces the concepts of knowledge representation and grammar and how does language relates to thought.
7. **Control Theory and cybernetics:** It introduces the concepts of designing the system that maximizes an objective function over time. This is roughly similar to the concepts of Al that behave optimally. It describes how artifacts (objects) can operate under their own control. That is, it introduces the concept of a self-controlling machine.
8. **Computer science and engineering:** This component introduces the concept of hardware, software, and operating system. Apart from this, it also discusses the programming language and tools used in Al.

Uses/Applications of Al

The potential applications of Artificial Intelligence are abundant (plentiful). They stretch from the military for autonomous control and target identification, to the entertainment industry for computer games and robotic pets. Let's also not forget big establishments dealing with huge amounts of information such as hospitals, banks, industries, and insurances, which can use Al to predict customer behavior and detect trends.

1. **Game playing:**

General game playing (GGP) and General video game playing (GVGP) is the concept and designs for artificial intelligence programs to successfully play plenty of games. For video games, game rules have to be either learned over multiple repetitions by artificial players or are predefined manually in a domain-specific language and sent in advance to artificial players. For instance, the GGP of chess, computers are programmed to play these games using a specially designed algorithm. It was considered a necessary landmark on the way to Artificial General Intelligence. The first commercial practice of general game-playing technology was Zillions of Games in 1998.

2. **Speech recognition:**

In speech recognition, the input is given to the computer in the form of vibrations produced by the sound. This is done with the help of an analog to digital converter that converts the vibrations produced by the sound into digital format.

Then, a set of complex algorithms runs on that data to recognize the speech and return a text as a result. Depending upon the goal, the end result may vary to some extent. For example, Google Voice typing converts spoken words into suitable text format while personal assistants like Siri and Google Assistant take the sound as input and convert it into both voice and text format, giving output as per the user's requirement.

3. **Understanding natural language:**

Natural language understanding is a branch of artificial intelligence that uses computer software to take the input in the form of sentences using text or speech. It simply reduces the gap between humans and computers allowing them to interact easily with each other.

4. **Computer vision:**

Computer vision is a field of artificial intelligence (AI), which enables the computer and its systems to get input in the form of digital images and videos and take action based on the provided input.

5. **Expert systems:**

An expert system is a computer system that mimics or even surpasses the decision-making ability of a human expert. It is generally designed to solve complex problems by surfing through bodies of knowledge. It is further divided into two subsystems; the knowledge base (which represents facts and rules) and inference engine (which applies the rules to the known facts to deduce new facts).

6. **Robotics:**

Artificial intelligence (AI) in robotics is the ability of the computer or the robot to perform multiple tasks performed by humans, which require human intelligence and discernment. It gives robots a computer vision to navigate, sense, and calculate their reaction accordingly For example: Robotic packaging uses various forms of Al for quicker and

accurate packaging at a lower price. Likewise, Sophia which is also marked as a "social robot" is successfully able to mimic social behavior and induce feelings of love in humans.

7. **Theorem proving:**

Proving theorems requires high intelligence as many of the practical problems can be cast in terms of theorems. If knowledge is expressed by logic, proving theorem is reasoning. It uses various AI techniques such as heuristic search.

8. **Symbolic mathematics:**

Symbolic mathematics refers to the manipulation of formulas, rather than doing arithmetic on numeric values. It is often used in conjunction with ordinary scientific computation as a generator of programs, used to actually do the calculations.

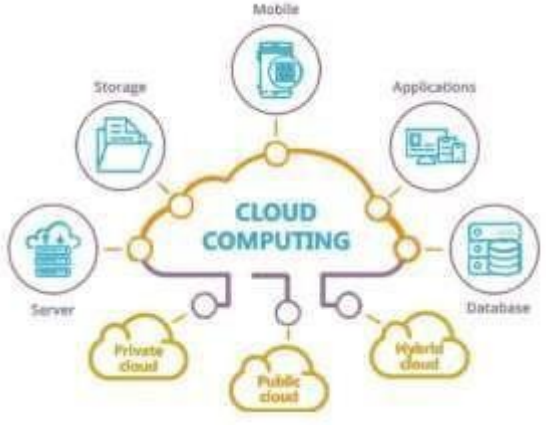
Robotics

Robotics is the branch of technology that deals with the design, construction, operation, and application of robots. It is a discipline overlapping artificial intelligence and mechanical engineering. It is concerned with building robot programmable devices consisting of mechanical actuators and sensory organs that are linked to a computer. The mechanical structure might involve manipulators, as in industrial robotics, might concern the movement of the robot as a vehicle, as in mobile - robotics. Robotics research is used in artificial intelligence as a framework for exploring key problems and techniques through a well-defined application.

1. Robotics is the branch of technology that deals with the design, construction, operation, and application of robots.
2. It is a discipline overlapping artificial intelligence and mechanical engineering.
3. It is concerned with building robots programmable devices consisting of mechanical actuators and sensory organs that are linked to a computer.
4. Robots are being used in applications like: Industry, scientific research, Military applications, Intelligent home applications, Health Services.

Cloud Computing

Cloud computing is the use of various services, such as software development platforms, servers, storage, and software, over the Internet, often referred to as the "cloud". It is defined as a type of computing that relies on sharing computing resources rather than having handle applications.



In cloud computing, the word cloud is used to represent "the Internet," so the phrase cloud computing means "a type of Internet-based computing," where different services - such as servers, storage, and applications are delivered to an organization's computers and devices through the Internet. Cloud computing allows application software to be operated using internet-enabled devices.

Types of Clouds

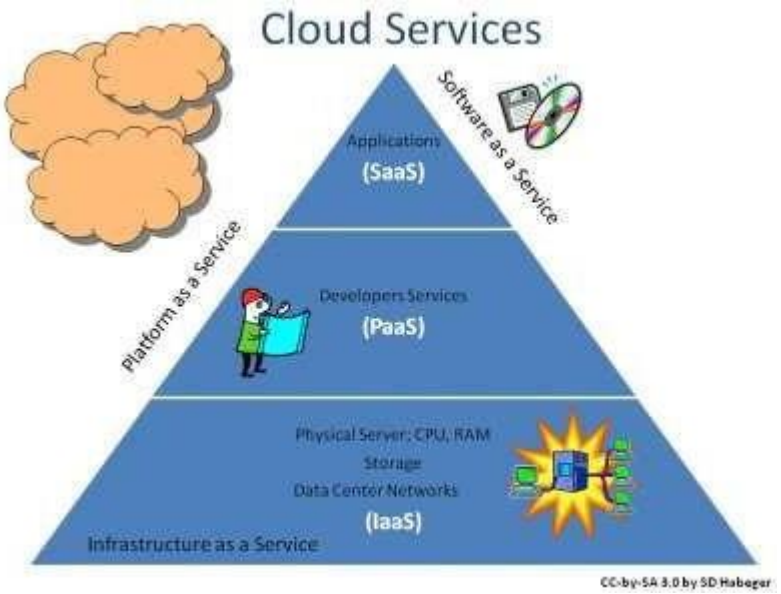
Clouds can be classified as ***public, private, and hybrid***. **Public cloud** is made available to the general public or a large industry group. **Private cloud** computing environment resides within the boundaries of an organization and is used exclusively for the organizational benefits **Hybrid cloud** is the combination of both public and private cloud. Sensitive With this cloud organizations might run non-core applications in a public cloud, while maintaining core applications and data in a private cloud.

Service Models of Cloud Computing

1. IaaS (Infrastructure as a Service): In this service, computing infrastructural components like server hardware, storage, bandwidth, and other fundamental computing resources are provided through the cloud.

2 Saas (Software as a Service): This service includes complete software on the can access software hosted on the cloud without installing it on the user's own computer.

3. PaaS (Platform as a Service): It allows the user to rent virtualized servers and associated services used to run existing applications, or to design, develop, test, deploy and host applications. It provides clients with access to the basic operating software and optional services to develop and use software applications without the need to buy and manage the underlying computing infrastructure.



Advantages of Cloud Computing

Some of the advantages of this technology are:

1. **Cost-efficient:** It is probably the most efficient method to use, maintain and upgrade.
2. **Almost unlimited storage:** Storing information in the cloud gives us almost unlimited storage capacity.
3. **Backup and recovery:** Since, all the data is stored in the cloud, backing it up and restoring the same is relatively much easier than storing the same on a physical device.

4. **Automatic software integration:** In the cloud, software integration is usually something that occurs automatically. It also allows us to customize the options with great ease.
5. **Easy access to information:** Once the user is registered in the cloud, the user can access the information from anywhere, where there is an Internet connection.
6. **Quick deployment:** Once the method of functioning is selected, the entire system can be fully functional in a matter of few minutes.

Disadvantages of Cloud Computing

Despite its many benefits, as mentioned above, cloud computing also has its disadvantages.

1. Technical issues: This technology is always prone to outages and other technical issues. Even the best cloud service providers run into this kind of trouble. Despite keeping up high standards of maintenance.

2. Security in the cloud: Storing all the sensitive information to a third-party cloud service provider could potentially put the company at great risk.

3. Prone to Attack: Storing information in the cloud could make the company vulnerable to external threats and attacks.

Big Data

Big Data refers to complex and large data sets that have to be processed and analyzed to uncover valuable information that can benefit businesses and organizations.

It has features like:

1. It refers to a massive amount of data that keeps on growing exponentially with time.
2. It is so voluminous that it cannot be processed or analyzed using conventional data processing techniques.
3. It includes data mining, data storage, data analysis, data sharing, and data visualization.
4. The term is an all-comprehensive one including data, data frameworks, along the tools and techniques used to process and analyze the data.

According to **Gartner**, the definition of Big Data- "Big data is high-volume, velocity, and information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making."

Types of Big Data

Big data can be classified as Structured, unstructured, and semi-structured.

1. Structured: It means that data can be processed, stored, and retrieved in a fixed format. It refers to highly organized information that can be readily and seamlessly stored and accessed from a database by simple search engine algorithms.

2. Unstructured: It refers to the data that lacks any specific form or structure whatsoever. This makes it very difficult and time-consuming to process and analyze unstructured data.

3. Semi-structured: It relates to the data containing both the formats mentioned above, that is structured and unstructured data. To be precise, it refers to the data that although

has not been classified under a particular repository (database), yet contains vital information or tags that segregate individual elements within the data.

Characteristics of Big Data

The main characteristics of big data are:

1. Variety: It refers to the variety of data gathered from multiple sources. The variety can be structured, unstructured, or semi-structured.

2. Velocity: It refers to the speed at which data is being created in real-time. It also comprises the rate of change, linking of incoming data sets at varying speeds, and activity bursts.

3. Volume: Big Data indicates huge 'volumes of data that are being generated daily from various sources like social media platforms, business processes, machines, networks, human interactions, etc.

4. Veracity: It refers to the reliability or trustworthiness of the data. Due to the large volume of data, we have uncertainty about the validity, the accurateness of data.

5. Value: It refers to the worth of business value of the collected data.

6. Variability: It refers to the inconsistency of the big data and how the big data can be used and formatted.

Application Areas of Big Data

Major application of big is data is:

1. Healthcare or Medical sector.
2. Academia.
3. Banking.
4. Manufacturing.
5. Information Technology (IT).
6. Retail business.
7. Transportation.

Advantages of Big Data Processing

Some of the advantages of big data processing are:

1. Businesses can utilize outside intelligence while taking decisions.
2. Improved customer service.
3. Early identification of risk to the product/services,
4. Better operational efficiency.
5. Big data analysis derives innovative solutions. It helps in understanding and targeting customers. It helps in optimizing business processes.

Disadvantages of Big Data Processing

Despite its many benefits, big data processing has the following disadvantages.

1. Traditional storage can cost a lot of money to store big data.
2. Big data analysis is not useful in the short run. It needs to be analyzed for a longer duration to leverage its benefits.
3. Big data analysis results are sometimes misleading.

Virtual Reality

Virtual reality (VR) is a term that expresses computer-based simulated environments. Which can perceive as in the real world, as well as in unreal worlds.

The virtual reality environments are primarily concerned with the 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 or headphones.

Virtual reality creates such a realistic artificial environment that the s/he should feel as in the real world. Today the Virtual reality (VR) technology is applied to advance fields of medicine, engineering, education, design, training, and entertainment.

Some of the application areas of virtual reality are:

1. It can be used in medical studies to enable students to know the human body.
2. It can be used in scientific research laboratories so that scientists can easily research a structure.
3. It can be used in entertainment like games and movies to make the gaming experience more real and to allow individuals to experience adventures under extreme conditions.
4. It can be used in driving schools as it gives a real look at roads and traffic.
5. It can be used in military training for the soldiers to get familiar with different areas on the battlefield.

Advantages of Virtual Reality

Some of the advantages of virtual reality are:

1. Virtual reality creates a realistic world.
2. It enables users to explore places.
3. Through Virtual Reality, users can experiment with an artificial environment.
4. Virtual Reality makes education easier and more comfortable.

Disadvantages of Virtual Reality

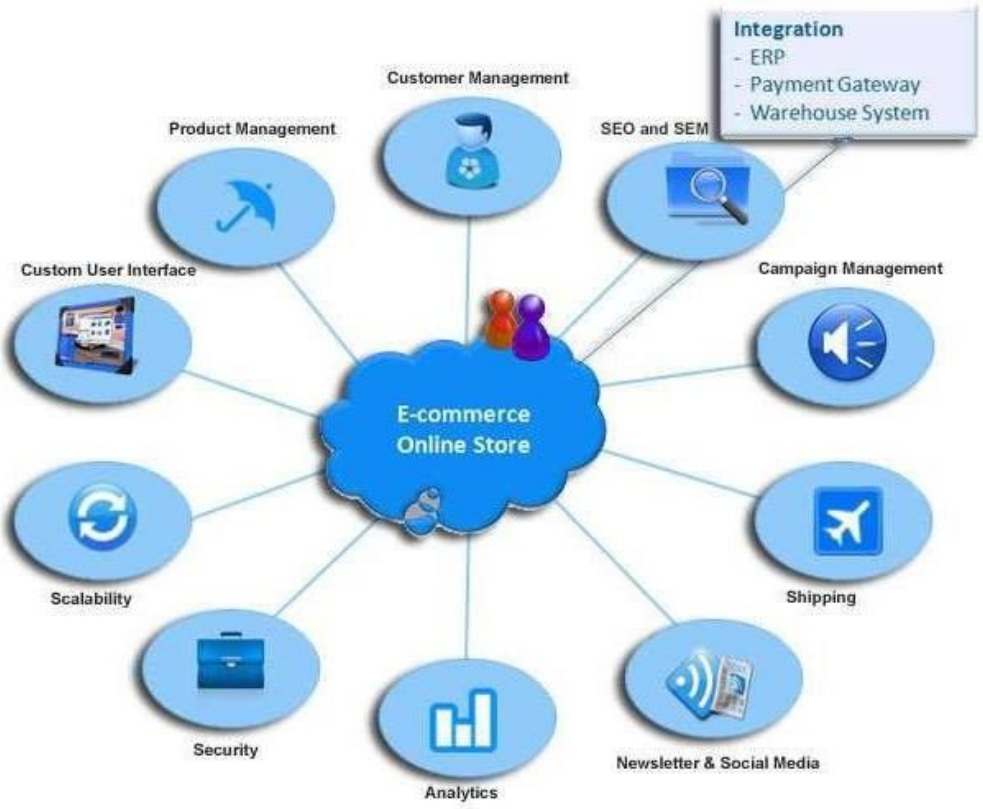
Some of the disadvantages of virtual reality are:

1. The equipment's used in virtual reality are very expensive.
2. It consists of complex technology.
3. In virtual reality environment we can't move by our own like in the real world.

e-Commerce, e-Medicine, e-Governance

e-Commerce

Electronic commerce (e-Commerce) is a process of buying and selling or exchanging products, services, and information using electronic media. There are many definitions for electronic commerce that include elements of electronic transactions and the buying and selling of goods and services online.



e-Commerce is a modern business methodology that addresses the needs of organizations, merchants, and consumers to cut costs while improving the quality of manufactured goods, services and increasing the speed of service delivery.
More commonly, e-commerce is associated with the buying and selling of products, and services via computer networks,
The main platforms of e-commerce remain the Internet, e-mail, fax, telephone orders

Classification of e-Commerce

1. B2B (Business to Business) Sells products or services to other businesses. e.g. www.freemarkets.com
2. B2C (Business to Consumer) Sells products or services directly to consumers. eg.. www.amazon.com, www.yahoo.com.
3. C2B (Consumer to Business) Consumer fixes a price on their own, which businesses accept or decline, e.g., www.priceline.com
4. C2C (Consumer to Consumer) Consumer sells directly to other consumer. e.g. www.ebay.com

Advantage of e-Commerce

Some of the advantages of e-commerce are:

1. It enables more individuals to work at home, and to do less traveling for shopping, resulting in less traffic on the roads, and lower air pollution.
2. It allows some merchandise to be sold at lower prices, benefiting less affluent people.
3. It enables people in Third World countries and rural areas to enjoy products and services which otherwise are not available to them.
4. Facilitates delivery of public services at a reduced cost, increases effectiveness, and/or improves quality.
5. It enables consumers to shop or do other transactions 24 hours a day, all year round from almost any location.
6. It provides consumers with more selections or choices.
7. It provides consumers with less expensive products and services by allowing them to shop in many places and conduct quick comparisons.
8. It allows quick delivery of products and services, especially with digitized products.
9. Consumers can receive relevant and detailed information in seconds, rather than in days or weeks walk-around to search a product.
10. It makes it possible to participate in virtual auctions. It allows consumers to interact with other consumers in electronic communities and exchange ideas as well as compare price-tag.
11. It facilitates competition, as a result of substantial discounts.
12. It expands the marketplace to national and international markets. It decreases the cost of creating processing, distributing, storing, and retrieving paper based information.

Disadvantage of e-Commerce

1. Businesses often calculate return on investment numbers before committing to any new technology. Costs, which are a function of technology, can change dramatically during even short-lived e-commerce implementation projects.
2. Many companies have had trouble recruiting and retaining employees with the technological, design, and business process skills needed to create an effective e-commerce presence.
3. The difficulty of integrating existing databases and transaction-processing software designed for traditional commerce into the software that enables e-commerce.
4. Many businesses face cultural and legal impediment (barrier) to e-commerce. Some consumers are still fearful (afraid) of sending their credit card numbers over the Internet.
5. Consumers are simply resistant to change and are uncomfortable viewing merchandise on a computer screen rather than in person.

e-Medicine

e-Medicine is an online clinical medical knowledge database, which is an approach to providing health care service to a large number of people spread in different locations.

It is mainly beneficial for the people of rural areas with limited or no medical facilities. e-Medicine is targeted to provide high-quality healthcare service. It minimizes the time and cost required for treatment.

e-Medicine usually contains up-to-date, searchable, peer-reviewed medical journals, online physician reference textbooks, and a complete article database on medical specialties. This Internet medical library and clinical knowledge base are available to physicians, medical students, nurses, other health professionals, and patients.

With the use of e-Medicine, doctors and patients who are physically apart can connect so that patients can share his/her problem with the doctor, and the doctor can suggest treatment or any test required.

e-Governance

e-Governance is the use of information and communication technology (ICT) to enhance the access and delivery of government services to benefit citizens, business partners, and employees. It transforms the traditional government using ICT to make it clear, effective, and accountable. However, it doesn't mean that putting more computers on the desks of government officials is e governance.

Governance is more than just a government website on the Internet. Political, social, economic, and technological aspects determine e-governance. It establishes a relationship between government officials and citizens, providing greater access to government information and services by making the government accessible online, promoting citizen participation by enabling citizens to interact more conveniently with government officials, such as by requesting government service and filing required documents through the website, increasing government accountability by making its operations more transparent, thereby reducing the opportunities for corruption, and supporting development goals by providing business, rural and traditionally underserved communities with information, opportunities, and communications capabilities. For example,

For example,

https://www.nepal.gov.np/, https://www.moe.gov.np/, https://www.moha.gov.np/

Objectives of e-Governance

Some of the objectives of e-Governance are:

- E-Governance refers to the provision of online public services to citizens and businesses.
- Services for citizens include the registration to government services such as health care, education, or employment benefits.
- For businesses, E-Governance services can take the form of online alerts for public procurements or funding opportunities as well as information and support on applicable legislation in a given sector.
- E-Governance helps to cut down their administrative costs, speed up procedures and therefore increase efficiency and reactivity.
- It could improve and accelerate administrative efficiency.

Challenges of implementing e-Governance

The key challenges of implementing E-Governance mainly in developing countries like Nepal are

- High-speed infrastructure to access the Internet is required.
- Creating trust and transparency of successful delivery of E-Governance service.
- The digital divide exists in developing countries. All the citizens may not have ICT knowledge.
- Network security and protection against viruses, spam, unwanted attacks, etc.
- Online privacy.
- All the citizens may not have access to computing resources.

7.6 Mobile Computing

Mobile computing is a generic term describing one's ability to use technology while moving as opposed to portable which is only practical for use while deployed in a stationary configuration. A mobile computing device is created using mobile components, such as mobile hardware and software. Mobile computing devices are portable devices capable of operating executing, providing services and applications like a computing device. It is a computing device used in transit. Users can access data and information from wherever they are.

Many types of mobile computers have been introduced since the 1990s, including a wearable computer, PDA, enterprise digital assistant, smartphone, UMPC (Ultra-mobile PC), Tablet PC

Features of Mobile Computing Device

Features of Mobile Computing devices are

- It is a portable device that can be used during mobility.
- It has limited processing and storage capability.
- It includes mobile communication, mobile hardware, and mobile software.
- It usually contains a touch screen for providing input.
- It contains an on-screen or virtual keyboard for proving text inputs. However, an external keyboard can be connected by using the USB port, infrared, or Bluetooth.
- It contains a camera, speaker, and microphone.
- It contains handwriting recognizing software.
- Most mobile computing devices contain a memory card slot to expand the storage capacity.
- It has wireless connectivity such as Bluetooth, Wi-Fi to connect the Internet or with other computing devices as well as a wired connection through the USB port connectivity services like need either Wi-Fi
- The most mobile computing device can synchronize their data with applications on users' computers.
- It can be used for cloud computing and remote access.
- It uses a mobile computing operating system such as Android, iOS, Windows Mobile OS, Palm OS.
- It can include GPS (Global Positioning System) receiver for navigation.

Advantages of Mobile Computing

Advantages of mobile technology are:

- It enables users to work from any location at any time.
- It saves time for accessing data and information.
- It helps to increase the productivity of users reducing the time and cost.
- It has made research easier.
- It is one of the major handheld sources of entertainment of users at present.
- Nowadays, Business processes are easily available through secured mobile connections.
- It is portable.
- It supports cloud computing.
- It provides remote access to the organizational data from any location.
- It is an independent platform. It can be accessed from any hardware or software.

Disadvantages of Mobile Technology

- Mobile technology requires faster and quality or GPRS or 3G or 4G connectivity.
- It has security concerns; most wireless connectivity is unsafe.
- Large power consumption is due to the use of batteries continuously and they do not tend to last long.
- The danger of misrepresentation i.e., credential
- Extensive use of mobile devices results in health problems.

7.7 Internet of Things (IoT)

Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity, which enables these things to connect, collect and exchange data.

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals, or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human to-computer interaction.

By combining these connected devices with automated systems, it is possible to "gather information, analyse it and create an action" to help someone with a particular task or learn from a process. A thing in the internet of things can be a person with a heart monitor implant, an animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low, or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and can transfer data over a network.

Advantages of IoT

- It automates tasks and helps to improve the quality of a business's services and reduces.
- It helps to operate the business operations more efficiently, better understand customers to deliver enhanced customer service.
- It supports to improve decision-making and increases the value of the business.
- It has the ability to access information from anywhere at any time on any device.

- It provides improved communication between connected electronic devices.
- Transferring data packets over a connected network saves time, effort, and money.

Disadvantages of IoT

- As the number of connected devices increases and more information is shared between devices, the chances of the system being attacked also increases.
- Organizations may eventually have to deal with massive numbers (maybe even millions) of loT devices, and collecting and managing the data from all those devices will be challenging.
- If there's a bug in the system, every connected device will likely become corrupted.
- Since there's no international standard of compatibility for loT, it's difficult for devices from different manufacturers to communicate with each other.

7.8 e-Learning

e-Learning applies to a learning/teaching or understanding about a topic with the help of Information and Communication Technology. e-Learning allows us to learn anywhere and usually at any time, as long as we have a properly configured computer, networks, devices, etc. e-Learning can be CD ROM-based, Network-based, Intranet-based, or Internet-based.

It can include text, video, audio, animation, and virtual environments. It can be a very rich learning experience that can even go beyond the lecture-based crowded classroom. It's a self paced, hands-on learning experience. The quality of the electronic-based training, as in every form of training, is in its content and its delivery. However, e-learning can suffer from many of the same pitfalls (drawbacks) as classroom training, such as boring slides, monotonous speech, and little opportunity for interaction. The beauty of e-learning is that new software that allows the creation of very effective learning environments that can overcome the classic material being used in traditional learning. For example, http://www.howstuffworks.com/

The concept of e-learning has become more popular throughout the globe because of the Covid 19 pandemic. The tools like Zoom, Microsoft Teams, Cisco Webex Meetings, Google Meet are also used for learning purposes.

7.9 m-Commerce

m-Commerce (mobile commerce) is the buying and selling of goods and services through wireless technology i.e., handheld devices such as cellular telephones and personal digital assistants (PDAs).

Industries affected by m-commerce include:

- Financial services, including mobile banking (when customers use their handheld devices to access their accounts and pay their bills), as well as brokerage services (in which stock quotes can be displayed and trading conducted from the same handheld device).

- Telecommunications, in which service changes, bill payment, and account reviews can all be conducted from the same handheld device.
- Service/retail as consumers is given the ability to place and pay for orders on the fly.
- Information services, which include the delivery of entertainment, financial news, sports figures, and traffic updates to a single mobile device.

7.10 Social Media

Social Media is a computer-based technology that is used for the creation and sharing of information, ideas, interests, and other forms of expression via virtual communities and networks. Facebook, Twitter, YouTube are popular social media tools.

Advantages of Social Media

- It provides easier and faster way to communicate.
- It provides worldwide real-time sharing of news and educational content.
- It is one of the effective marketing/advertising tools at present.
- It is the major source of entertainment at present.
- It helps to understand better the latest trends and events.

Disadvantages of Social Media

- It has increased cyber-crime.
- Productive times is lost due to time waster in social media.
- It is a common tool at present for spreading rumours and fake news/updates.
- It has a high risk of fraud.
- It has decreased privacy.

Web References:

- https://www.javatpoint.com
- https://www.w3schools.com
- https://www.tutorialspoint.com
- https://www.google.com
- https://www.wikipedia.org