# ❖ CSC 103 - Fundamentals Of Computers And Applications

- **What is Computer?**
  - Computer is an electronic device. It accepts raw data from us, process it and gives meaningful information as required by user as per given instruction.
  - **Characteristics of computer:**
    - ✓ Computer is very fast and accurate
    - ✓ Works at constant efficiency
    - ✓ Can store information
    - ✓ Computer is very versatile

- **What are the classification of computer?**

  **Classification of computer**

  1. **Microcomputer**

     -Two types of microcomputer

     a. Home computer

     b. Business computer
  2. **Minicomputer**

     -Multi user, multi-tasking and faster than microcomputer

     -larger memory capacity, larger C.P.U
  3. **Main Frame**
     - Faster than mini computer
  4. **Super Computer**
     - Have great processing power

- **Computer System**

  Computer system consists of –

  1. **Hardware**- It consists of the mechanical, electrical, and electronics parts of the computer
  2. **Software**- A set of programs that takes input, process data & produce output
  3. **Firmware**- The programs that are stored in ROM (Read Only Memory)

- **Classification of Hardware**

Hardware classified as –

1. CPU (Central Processing Unit)
2. Peripherals

CPU classified as-

1. **CU (Control Unit)** - controls and directs the operations of entire computer system.
2. **ALU (Arithmetic & Logic Unit**) - Carries out arithmetic operations like addition, subtraction etc.
3. **Main Memory**

   **-** Classified as

   **a. RAM (Random Access Memory)-** Temporary memory used for storing data or instructions during processing

   **b. ROM (Read Only Memory)-** Permanent memory which contains basic information the CPU needs when you first turn on the computer as well as other times during the operation of the computer.

Peripherals can be classified as-

1. **Input**
2. **Output**
3. **Secondary storage**

- **Types of Software**
  1. **System Software**
     - System software is general purpose software which is used to operate computer hardware. It provides platform to run application software.
     - Example: compiler, assembler, debugger, driver, etc.
  2. **Application Software**
     - Application software is specific purpose software which is used by user for performing specific task.
     - Example: word processor, web browser, media player, etc.

- **What is Cache Memory?**

  Cache Memory is smaller and faster Memory which is very nearest to the CPU , all the recent Instructions are Stored into the Cache Memory.

- **What are the network topologies?**

  **Network Topology**

  A network topology is the arrangement of a network, including its nodes and connecting lines.

  Types-

  1. Bus Topology
  2. Star Topology
  3. Ring Topology
  4. Mesh Topology
  5. Tree Topology

1. In the **bus network topology**, every workstation is connected to a main cable called the bus. Therefore, in effect, each workstation is directly connected to every other workstation in the network.

2. In the **star network topology**, there is a central computer or server to which all the workstations are directly connected. Every workstation is indirectly connected to every other through the central computer.

3. In the **ring network topology**, the workstations are connected in a closed loop configuration. Adjacent pairs of workstations are directly connected. Other pairs of workstations are indirectly connected, the data passing through one or more intermediate nodes.

4. The **mesh network topology** employs either of two schemes, called full mesh and partial mesh. In the full mesh topology, each workstation is connected directly to each of the others. In the partial mesh topology, some workstations are connected to all the others,

and some are connected only to those other nodes with which they exchange the most data.

5. The **tree network topology** uses two or more star networks connected together. The central computers of the star networks are connected to a main bus. Thus, a tree network is a bus network of star networks.

## ❖ CSC 183 - Programming C

- **What is Identifier?**

  Identifiers are user defined word used to name of entities like variables, arrays, functions, structures etc. Rules for naming identifiers are:

  1) Name should only consist of alphabets (both upper and lower case), digits and underscore (_) sign.

  2) First characters should be alphabet or underscore

  3) Name should not be a keyword

  4) Since C is a case sensitive, the upper case and lower case considered differently, for example code, Code, CODE etc. are different identifiers.

  5) Identifiers are generally given in some meaningful name such as value, net_salary, age, data etc. An identifier name may be long, some implementation recognizes only first eight characters, most recognize 31 characters. ANSI standard compiler recognizes 31 characters. Some invalid identifiers are 5cb, int, res#, avg no etc.

- **Data types**
  1. Int
  2. Float
  3. Double
  4. Char
  5. Pointer
  6. Array
  7. Void

- **What is constant?**

  **Constant**

  Constant is an any value that cannot be changed during program execution.

- **What is variable?**

  **Variable**

  Variable is a data name which is used to store some data value or symbolic names for storing program. Declaration of variables specifies its name, data types and range of the value that variables can store depends upon its data types.

  Syntax:

  ```
   int a;
  char c;
   float f
  ```

- **Expressions**

  An expression is a combination of variables, constants, operators and function call. It can be arithmetic, logical and relational for example:-

  ```
   int z= x+y // arithmatic expression
  a>b //relational
  a==b // logical func(a, b) // function call
  ```

- **Operators**

  1. **Arithmetic Operator**

     This operator used for numeric calculation. These are of either Unary arithmetic operator, Binary arithmetic operator. Where Unary arithmetic operator required 25 *under revision only one operand such as +,-, ++, --,!, tiled. And these operators are addition, subtraction, multiplication, division.

  2. **Assignment Operator**

     A value can be stored in a variable with the use of assignment operator. The assignment operator(=) is used in assignment statement and assignment expression.

     For example,

```
int x= y;
int Sum=x+y+z;
```

3. **Relational Operator**

   It is use to compared value of two expressions depending on their relation. Expression that contain relational operator is called relational expression. Here the value is assign according to true or false value.

   a.(a>=b) || (b>20)

   b.(b>a) && (e>b)

   c. 0(b!=7)

- **Control Statement**

  Control statement defined How the control is transferred from one part to the other part of the program. There are several control statement like if...else, switch, while, do....while, for loop, break etc.

- **Loops in C**

  -It is a block of statement that performs set of instructions.

  There are three types of loops in c

  1.While loop

  2.do while loop

  3.for loop While loop

- **Array**

  Array is the collection of similar data types or collection of similar entity stored in contiguous memory location. Array of character is a string.

  Example: int arr[100];

  **Declaration of an array :**

  Its syntax is :

  Data type array name [size];

  int arr[100];

  int mark[100];

```
int a[5]={10,20,30,100,5}
```

### Two dimensional arrays

Two dimensional array is known as matrix.

Example:- int a[2][3];

### Function

A function is a self-contained block of codes or sub programs with a set of statements that perform some specific task or coherent task when it is called.

### Local, Global and Static variable

**Local variable:-**

variables that are defined with in a body of function or block. The local variables can be used only in that function or block in which they are declared. Same variables may be used in different functions such as

```
 function()
 { int a,b; function 1(); }
 function2 () { int a=0; b=20; }
```

**Global variable:-**

the variables that are defined outside of the function is called global variable. All functions in the program can access and modify global variables. Global variables are automatically initialized at the time of initialization.

Example:

```
 #include void function(void);
void function1(void);
void function2(void);
int a, b=20; void main()
 { printf("inside main a=%d,b=%d \n",a,b);
function();
function1();
 function2(); }
 function()
{ Prinf("inside function a=%d,b=%d\n",a,b); }
 function 1()
```

```
prinf(“inside function a=%d,b=%d\n”,a,b);
}
 function 2()
{ prinf(“inside function a=%d,b=%d\n”,a,);
 }
```

**Static variables:**

static variables are declared by writing the key word static. –

syntax:-

```
static data type variable name;
static int a;
```

-the static variables initialized only once and it retain between the function call. If its variable is not initialized, then it is automatically initialized to zero.

Example:

```
void fun1(void);
 void fun2(void);
void main()
 { fun1();
fun2();
}
void fun1()
{
int a=10, static int b=2;
printf(“a=%d, b=%d”,a,b);
a++; b++;
 }
```

}

- **Recursion**

  When function calls itself (inside function body) again and again then it is called as recursive function.

  Ex:- /*calculate factorial of a no.using recursion*/

  ```
  int fact(int);
  void main()
  {
   int num; printf(“enter a number”);
  scanf(“%d”,&num); f=fact(num);
  printf(“factorial is =%d\n”f);
  }
  fact (int num)
   {
  If (num==0||num==1)
  return 1;
  else
  return(num*fact(num-1));
  }
  ```

- **Pointer**

  A pointer is a variable that store memory address or that contains address of another variable where addresses are the location number always contains whole number.

  Syntax-

  ```
   Data type *pointer name;
  ```

  Example:

  ```
  void main()
   { int i=105;
  int *p;
  p=&i;
  printf(“value of i=%d”,*p);
  printf(“value of i=%d”,*/&i);
  ```

}

# ❖ CSC 283 - Programming C++

- **What is Object Oriented Programming Language?**
  Object-oriented programming is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.

- **What is Class?**

A class is a construct that enables you to create your own custom types by grouping together

variables of other types, methods and events.

A class is like a blueprint. It defines the data and behavior of a type.

- **What is member function?**

  A member function of a class is a function that has its definition or its prototype within the class definition similar to any other variable. It operates on any object of the class of which it is a member, and has access to all the members of a class for that object.

- **What is object?**

  An object is a specific instance of a class; it contains real values instead of variables.

- **What is encapsulation?**
  Grouping related data and functions together as objects and defining an interface to those objects.

- **What is inheritance?**
  Allowing code to be reused between related types.

- **What is polymorphism?**

  Allowing a value to be one of several types, and determining at runtime which functions to call on it based on its type.

- **Basic Structure:**

```
#include <iostream>
int main()
{
  cout << "Hello, world!\n";
  return 0;
 }
```

- **What is the access modifier?**

  The access restriction to the class members is specified by the labeled **public, private,** and **protected** sections within the class body. The keywords public, private, and protected are called access specifiers.

  A **public** member is accessible from anywhere outside the class but within a program.

  A **private** member variable or function cannot be accessed, or even viewed from outside the class. Only the class and friend functions can access private members.

  A **protected** member variable or function is very similar to a private member but it provided one additional benefit that they can be accessed in child classes which are called derived classes.

- **What is Constructor & Destructor?**

  **Constructor** is a special member function of class and it is used to initialize the objects of its class. It is treated as a special member function because its name is the same as the class name.

  **Destructor** on the other hand is used to destroy the class object.

- **What is Function Overloading?**

  Function overloading is a programming concept that allows programmers to define two or more functions with the same name and in the same scope. Each function has a unique signature which is derived from: function/procedure name, number of arguments.

- **What is Multiple Inheritance?**

  Multiple inheritance is a feature of some object-oriented computer programming languages in which an object or class can inherit characteristics and features from more than one parent object or parent class.

- **What is Exception Handling?**

Exceptions provide a way to transfer control from one part of a program to another. C++ exception handling is built upon three keywords: try, catch, and throw.

- throw − A program throws an exception when a problem sHows up. This is done using a throw keyword.
- catch − A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.
- try − A try block identifies a block of code for which particular exceptions will be activated. It's followed by one or more catch blocks.

# ❖ CSC 231 - Fundamentals of electronics and digital system

- **Atomic Structure**
  - Proton (positively charged particles found in the atomic nucleus)
  - Neutron (uncharged particles found in the atomic nucleus)
  - Electron(negatively charged particles that surround the atom's nucleus)

- **Electronic Materials**
  - Conductors: have low resistance which allows electrical current flow
  - Insulators: have high resistance which suppresses electrical current flow
  - Semiconductors: can allow or suppress electrical current flow. The main characteristic of a semiconductor element is that it has four electrons in its outer or valence orbit.

- **Crystal Lattice:** The unique capability of semiconductor atoms is their ability to link together to form a physical structure called a crystal lattice.

- **Types of Semiconductor**

1. **Intrinsic semiconductor**

An intrinsic semiconductor is a pure semiconductor which does not have any doping agent (an impurity added to crystal lattice).

2. **Extrinsic Semiconductors**

An extrinsic semiconductor is a semiconductor whose electrical conductivity can be increased by adding the trace amounts of other elements such as impurities in the material.

**Doping:** To make the semiconductor conduct electricity, other atoms called impurities must be added. “Impurities” are different elements. This process is called doping.

- **Types of Semiconductor Materials**

**N-type:** The silicon doped with extra electrons is called an “N type” semiconductor.

**P-type:** Silicon doped with material missing electrons that produce locations called holes is called “P type” semiconductor.

**Diode:** A diode is a specialized electronic component with two electrodes called the anode and the cathode.

- **Knee voltage/ Zener voltage / Breakdown voltage**

In the forward region, the voltage at which the current starts to increase rapidly is called the knee voltage of the diode. The knee voltage equals the barrier potential.

- $V_k \approx 0.7V$ (for silicon diode)
- $V_k \approx 0.3V$ (for Germanium diode)

- **Power dissipation:** The power rating is the maximum power the diode can safely dissipate without shortening its life or degrading its properties.

- **Barrier Voltage:** Barrier voltage prevents the further combination of electrons and holes which limits the size of depletion region.

- **Diode Biasing:** The voltage applied to the semiconductor diode is referred to as bias voltage.

Types of diode - Zener diode, LED (Light-emitting Diode), Photo diode etc.

- **Advantages of LEDs**
  - Low operating voltage, current and power consumption
  - LEDs have low inherent noise levels

- **Limitations of LEDs**
  - Temperature dependence of radiant output power and wavelength.
  - Sensitivity to damages by over voltage or over current.

- **Zener Regulator** : A zener diode is sometimes called a voltage-regulator diode because it maintains a constant output voltage even though the current through it changes.
- **Transistor:** Transistor is a three terminal electronic device .Transistor Amplifies the weak input signal. Transistor is used as switch and amplifier with various electronic devices.

A transistor consists of three layers of semiconductors:

1. Base 2. Emitter 3. Collector

The Bipolar Junction Transistor (BJT) is the simplest type of transistor. BJT are classified into two types:

- NPN Transistor
- PNP Transistor

- **Number System:** Number system can be defined as the method or format which is used for denoting a numerical value.

Types of number system:

- ➢ Decimal Number System – Consists of 10 digits starting from 0 to 9. Base is 10
- ➢ Binary Number System - Consists of 2 digits0, 1. Base is 2
- ➢ Octal Number System - Consists of 8 digits starting from 0 to7. Base is 8
- ➢ Hexadecimal Number System - Consists of 16 digits starting from 0 to 9, A- F. Base is 16.

- **BCD Code:** BCD stands for Binary Coded Decimal. It combines some of the characteristics of both the binary and decimal systems.

- **ASCII Code:** The American Standard Code for Information Interchange (ASCII) code is used to represent the 26 upper case letters (A to Z), 26 lowercase letters (a to z), 10 numbers (0 to 9), 33 special characters and symbols and 33 control characters.

- **E-ASCII Code:** E-ASCII stands for Extended ASCII. It is of 8-bits

- **UNICODE:** It provides a unique number for each and every character for every program, language and platform we use.

- **Analog Signals:** An analog signal is an AC or DC voltage or current that varies smoothly or continuously.

- **Digital signals:** Digital signals are essentially a series of pulses or rapidly changing voltage levels that vary in discrete steps or increments.

- **Digital Integrated Circuits (ICs):** A digital integrated circuit, also called as a chip, is a small electronic device made out of the semiconductor material that consists of electronic components like transistors, logic gates, flip-flops, multiplexers and other circuits.

  Types of IC:

  - SSI (Small-Scale Integration)
  - MSI (Medium-Scale Integration)
  - LSI (Large-Scale Integration)
  - VLSI (Very Large-Scale Integration)

- **Logic Circuit:** A collection of individual logic gates connect with each other and produce a logic design known as a Logic Circuit. The following are the three basic gates: NOT , AND , OR .

Universal gates are two gates namely NAND, NOR

- **Multiplexer:** Multiplex means many into one. A multiplexer is a circuit used to select and route any one of several input signals to a single output. The types of multiplexers are as follows:

- 2-to-1 Multiplexer
- 4-to-1 Multiplexer
- 8-to-1 Multiplexer
- 16-to-1 Multiplexer

- **Demultiplexer:** Demultiplex means one into many. A demultiplexer is a circuit with one input and many outputs. The types of multiplexers are as follows:
  - 1-to-2 Multiplexer
  - 1-to-4 Multiplexer
  - 1-to-8 Multiplexer
  - 1-to-16 Multiplexer

- **Encoder:** An encoder is a circuit with multiple inputs which generates a unique address at its output that is, it converts an active input signal to a coded output signal.

- **Decoder:** A decoder is a circuit with no data input, but accepts only the control inputs and generates many outputs

## ❖ CSC 197 - Assembly Language

- **What does Assembly Language mean?**

An assembly language is a low-level programming language for microprocessors and other programmable devices. It is not just a single language, but rather a group of languages. An assembly language implements a symbolic representation of the machine code needed to program a given CPU architecture.

Assembly language is also known as assembly code. The term is often also used synonymously with 2GL.Assembly language may also be called symbolic machine code.

- **What is Basic elements of assembly language?**

- Opcode mnemonics
- Data definitions
- Assembly directives

- **What is Microcomputer system?**

  A microcomputer system consists of a system unit, a keyboard, a display
  screen and a disk drivers
  The system unit is often referred to as the computer as it houses the circuit
  boards of the computer
  I/O devices perform input/output operations
  Peripheral devices or peripherals
  ICs are used to construct a computer circuit
  Contains thousands of transistors
  Digital circuits operate on voltage signal level
  High/low (1/0) voltage
- These symbols are called binary digits or bits
  All information represented by strings of 0's and 1's

- **Functionally computer circuits consist of three parts:**

  Central Processing Unit (CPU)

  Memory circuits

  I/O circuit

- **What is Memory?**

  Information processed by the computer is stored in its memory.

  A memory circuit element can store one bit of data.

  The memory circuits are usually organized into groups that can store eight bits of data.

  A string of eight bits is called a byte.

  Memory byte is identified by a number that is called its "address"

  Data stored in a memory byte are called its contents

  Example: Suppose a processor uses 20 bits for an address. How many memory bytes can be accessed?

  Solution: A bit can have two possible values, so in a 20-bits address there can be 220 = 1,048,576 different values, with each value being the potential address of a memory byte. In computer terminology, the number 220 is called 1 mega. Thus, a 20-bit address can be used to address 1 megabyte or 1 MB.

- **What is the bit position of Memory?**

  **Bit position**

  Represents in a microcomputer as word and a byte.

  Positions are numbered from right to left, starting with 0.

  In a word, 0 to 7 form as low byte and from 8 to 15 form as high byte

- **How memory operation work?**

  **Memory operations:** Processor can perform two operations on memory

  Read (fetch) – the content of a location, processor gets a copy of the data

  Write (store) – data at a location, the new contents of the location

  There are two kinds of **memory circuits**

  RAM

Locations can be read and written

Contents of RAM memory are lost when the machine is turned off

ROM

Location can only be read

ROM retains their values even when the power is off

- **What is Buses?**

  A set of wires or connections called buses that connect the different components

  There are three kind of buses:

  Address bus

  Data bus

  Control bus

- **What is The CPU?**

  Refers as the brain of the computer.

  Controls the computer by executing programs stored in memory.

  System program

  An application program

  Instructions performed by a CPU are called its instruction set

  Intel 8086 microprocessor as an example of a CPU

  There are two main components:

  Execution unit

  Bus interface unit

- **Execution Unit (EU)**

  The purpose of the EU is to execute instructions.

  Contains a circuit called the arithmetic and logic unit (ALU).

  ALU can perform arithmetic (+, - , / , x) and logic (AND, OR, NOT) operations.

  Data for the operations are stored in circuits called registers.

  A memory location except that normally refer to it by a name rather than a number.

EU has eight registers for storing data;

EU contains temporary ' registers for holding operands for the ALU, and the FLAGS register

- **Bus Interface Unit (BIU)**

Facilitates communication between the EU and the memory or I/O circuits.

Responsible for transmitting addresses, data, and control signals on the buses.

Registers hold addresses of memory locations.

Fetches up to six bytes of the next instruction and places them in the instruction queue

Instruction prefetch

- **What is I/O Ports?**

I/O devices are connected to the computer through I/O circuits

Contains several registers called I/O ports

I/O ports have addresses and are connected to the bus system.

Allows the CPU to distinguish between an I/O port and a memory location.

**Serial and Parallel Ports**

Data transfer between an I/O port and an I/O device can be **1 bit** at a time (serial), or 8 or 16 bits at a time (parallel)

- **How Instruction execution work?**

A machine instruction has two parts:

Opcode

Specifies the type of operation.

Operands

Provide memory addresses to the data to be operated on.

The CPU goes through the following steps to execute a machine instruction

Fetch

Execute

Fetch and Execute

1. Fetch an instruction from memory.
2. Decode the Instruction to determine the operation.
3. Fetch data from memory if necessary.
4. Perform the operation on the data.
5. Store the result in memory if needed.

- **Some of Programming Language**

Machine language

A CPU can only execute machine language instructions

Bit strings

Assembly language

A more convenient language

Uses symbolic names to represent operations, registers, and memory
locations.

Converts to machine language before the CPU can execute it

Assembler

- **Advantages of Assembly Languages**

-Produces a faster, shorter machine language program.

-Easy to read or write to specific memory locations and I/O ports.

-Many high-level languages accept subprograms written in assembly language.

-Assembly language program

- **What is Number systems?**

Decimal system /Positional number system

Each digit in the number is associated with a power of
10, according to its position in the number

Binary Number System

The base is two and there are only two digits, 0 and 1.

The binary string **11010** represents the number.

Hexadecimal Number System

Numbers written in binary tend to be long and difficult to express.

16 bits are needed to represent the contents of a memory word in an 8086-based computer.

Assembly language programs tend to use both binary, decimal, and a third number system

Hexadecimal

Conversion between binary and hex is easy

A base sixteen system.

Digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

Each hex digit corresponds to a unique four-bit number.

- **What is Unsigned Integers?**

An unsigned integer is an integer that represents a magnitude, so it is never negative addresses of memory locations, counters, and ASCII character codes. None of the bits are needed to represent the sign.The largest unsigned integer that can be stored in a byte is 11111111 = FFh = 255. The biggest unsigned integer a 16 bit word can hold is 1111111111111111 = FFFFh =65535.

- **What is Signed Integers?**

A signed integer can be positive or negative.

The most significant bit is reserved for the sign:

1 means negative and 0 means positive.

Negative integers are stored in a computer in a special way known as two's complement.

One's Complement

an integer is obtained by complementing each bit; that is, replace each 0 by a 1 and each 1 by a 0.

Find the one's complement of 5.

Converting to 16 bits binary, 0000000000000101

5 = 0000000000000101

One's complement of 5 = 111l111111111010

Find the one's complement of 18.

Converting to 16 bits binary, 0000000000010010

One's complement of 18 = 1111111111101101

Two's Complement

To get the two's complement of an integer, just add 1 to its one's complement.

Find the two's complement of 5.

- **What is Character Representation?**

  Not all data processed by the computer are treated as numbers

  the video monitor and printer are character oriented

  Characters must be coded in binary in order to be processed by the computer.

  ASCII code

  This system uses seven bits to code each character, so there are a total of 27= 128 ASCII codes.

  Only **95** ASCII codes, from 32 to 126, are considered to be printable.

  The codes 0 to 31 and also 127 were used for communication control purposes and **do not** produce printable characters.

  the control characters are used to control the operations of these devices.

- **What is registers?**

  Information inside the microprocessor is stored in registers.

   Classify according to the functions they perform.

   Data registers hold data for an operation.

  1. AX (Accumulator Register)

2. BX (Base Register)

3. CX (Count Register)

4. DX (Data Register)

□ Address registers hold the address of an instruction or data.

1. segment registers

- Code Segment

- Code Segment

- Code Segment

1. Index and PointerRegisters

- Instruction Pointer (IP)

- Stack Pointer (SP)

- Base Pointer (BP)

- Source Index (SI)

- Destination Index (DI)

□ A status registers keeps the current status of the processor.

□ A Flag registers

- Status flag

1. Carry Flag (CF)

2. Carry Flag (CF)

3. Auxiliary Carry Flag (AF)

4. Zero Flag (ZF)

5. Sign Flag (SF)

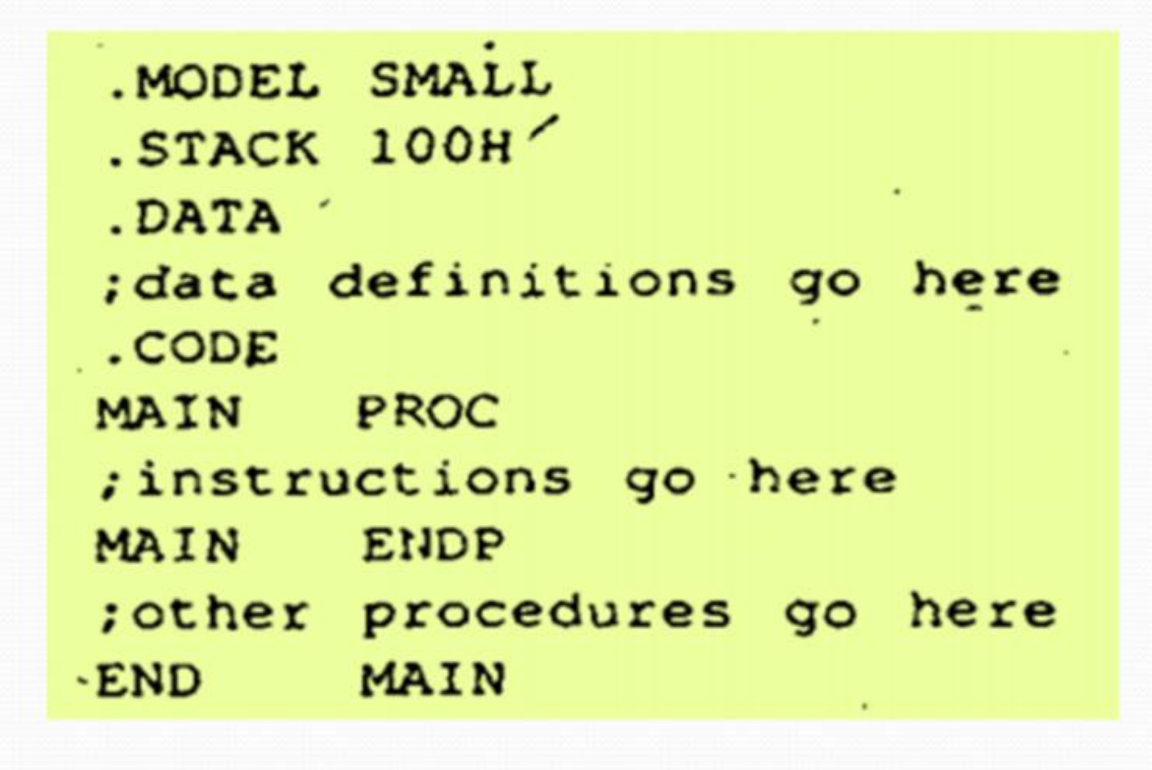
6. Overflow Flag (OF)

- control flag

1. trap flag
2. interrupt flag
3. Direction flag

- There are fourteen 16-bitregisters.

- **Assembly Struture**



```
 .MODEL SMALL
 .STACK 100H
 .DATA
;data definitions go here
 .CODE
MAIN    PROC
;instructions go here
MAIN    ENDP
;other procedures go here
END     MAIN
```

# ❖ CSC 391 – Data Structure & Algorithm

- **Data Structures and Algorithms**

Data Structure is a way of collecting and organising data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage. For example, we have some data which has, player's **name** "Virat" and **age** 26. Here "Virat" is of **String** data type and 26 is of **integer**data type.

- **Basic types of Data Structures**

As we have discussed above, anything that can store data can be called as a data structure, hence Integer, Float, Boolean, Char etc, all are data structures. They are known as **Primitive Data Structures**.

Then we also have some complex Data Structures, which are used to store large and connected data. Some example of **Abstract Data Structure** are :

- Linked List
- Tree
- Graph
- Stack, Queue etc.

- **Space Complexity**

It's the amount of memory space required by the algorithm, during the course of its execution. Space complexity must be taken seriously for multi-user systems and in situations where limited memory is available.

An algorithm generally requires space for following components :

**Instruction Space:** It's the space required to store the executable version of the program. This space is fixed, but varies depending upon the number of lines of code in the program.

**Data Space:** It's the space required to store all the constants and variables(including temporary variables) value.

**Environment Space:** It's the space required to store the environment information needed to resume the suspended function.

- **Time Complexity**

Time Complexity is a way to represent the amount of time required by the program to run till its completion. It's generally a good practice to try to keep the time required minimum, so that our algorithm completes its execution in the minimum time possible.

- **Tree Data Structure**

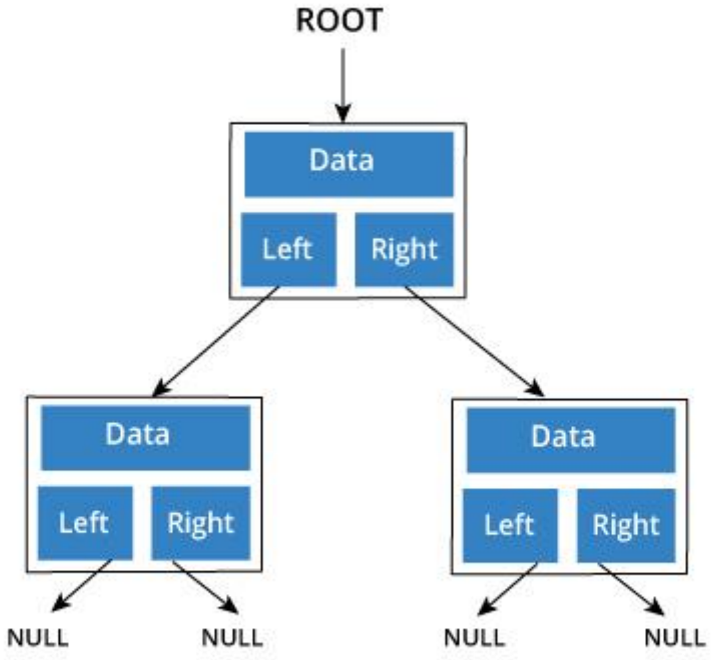
A linked list is a chain of nodes connect through "next" pointers. A tree is similar, but each node can be connected to multiple nodes.

When we talk about tree, mostly we mean binary tree, that is a structure that has two children, left and right.

- **Binary Tree Representation**

A node of a binary tree is represented by a structure containing a data part and two pointers to other structures of the same type.

```
struct node

{

  int data;

  struct node *left;

  struct node *right;

};
```



A special pointer called ROOT points to the node that is the parent of all the other nodes.

Also, the nodes that don't have any children have their left and right pointers point to NULL.

- **Hash Table**

Hash Table is a data structure which stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value.

- **Hashing**

Hashing is a technique to convert a range of key values into a range of indexes of an array.

- **Sorting**

Sorting is nothing but arranging the data in ascending or descending order. The term **sorting** came into picture, as humans realised the importance of searching quickly.

**Sorting** arranges data in a sequence which makes searching easier.

- **Sorting Efficiency**

Since the beginning of the programming age, computer scientists have been working on solving the problem of sorting by coming up with various different algorithms to sort data.

The two main criterias to judge which algorithm is better than the other have been:

1. Time taken to sort the given data.
2. Memory Space required to do so.

- **Bubble Sort Algorithm**

**Bubble Sort** is a simple algorithm which is used to sort a given set of n elements provided in form of an array with n number of elements. Bubble Sort compares all the element one by one and sort them based on their values.

- **Selection Sort Algorithm**

Selection sort is conceptually the most simplest sorting algorithm. This algorithm will first find the **smallest** element in the array and swap it with the element in the **first** position, then it will find the **second smallest** element and swap it with the element in the **second** position, and it will keep on doing this until the entire array is sorted.

- **Insertion Sort Algorithm**

Consider you have 10 cards out of a deck of cards in your hand. And they are sorted, or arranged in the ascending order of their numbers.

If I give you another card, and ask you to **insert** the card in just the right position,so that the cards in your hand are still sorted. What will you do?

Well, you will have to go through each card from the starting or the back and find the right position for the new card, comparing it's value with each card. Once you find the right position, you will **insert** the card there.

Similarly, if more new cards are provided to you, you can easily repeat the same process and insert the new cards and keep the cards sorted too.

This is exactly How **insertion sort** works. It starts from the index 1(not 0), and each index starting from index 1 is like a new card, that you have to place at the right position in the sorted subarray on the left.

- **Merge Sort Algorithm**

Merge Sort follows the rule of **Divide and Conquer** to sort a given set of numbers/elements, recursively, hence consuming less time.

- **Quick Sort Algorithm**

Quick Sort is also based on the concept of **Divide and Conquer**, just like merge sort. But in quick sort all the heavy lifting(major work) is done while **dividing** the array into subarrays, while in case of merge sort, all the real work happens during **merging** the subarrays. In case of quick sort, the combine step does absolutely nothing.

- **Heap Sort Algorithm**

Heap Sort is one of the best sorting methods being in-place and with no quadratic worst-case running time. Heap sort involves building a **Heap** data structure from the given array and then utilizing the Heap to sort the array.

- **What is Stack Data Structure?**

**Stack** is an abstract data type with a bounded(predefined) capacity. It is a simple data structure that allows adding and removing elements in a particular order. Every time an element is added, it goes on the **top** of the stack and the only element that can be removed is the element that is at the top of the stack, just like a pile of objects.

- **What is a Queue Data Structure?**

**Queue** is also an abstract data type or a linear data structure, just like stack data structure, in which the first element is inserted from one end called the **REAR**(also called **tail**), and the removal of existing element takes place from the other end called as **FRONT**(also called **head**).

This makes queue as **FIFO**(First in First Out) data structure, which means that element inserted first will be removed first.

- **What is a Circular Queue?**

Before we start to learn about Circular queue, we should first understand, why we need a circular queue, when we already have linear queue data structure.

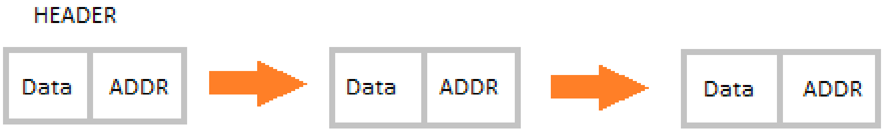
In a Linear queue, once the queue is completely full, it's not possible to insert more elements. Even if we dequeue the queue to remove some of the elements, until the queue is reset, no new elements can be inserted.

- **Linked Lists**

Linked List is a very commonly used linear data structure which consists of group of **nodes** in a sequence.

Each node holds its own **data** and the **address of the next node** hence forming a chain like structure.

Linked Lists are used to create trees and graphs.



- **Recursion Basics**

Some computer programming languages allow a module or function to call itself. This technique is known as recursion. In recursion, a function **α** either calls itself directly or calls a function **β** that in turn calls the original function **α**. The function **α** is called recursive function.

- **Algorithm in Programming**

1. **What is an Algorithm?**

   Ans: In programming, algorithm is a set of well defined instructions in sequence to solve the problem.

2. **Qualities of a good algorithm**.

   - Input and output should be defined precisely.
   - Each steps in algorithm should be clear and unambiguous.
   - Algorithm should be most effective among many different ways to solve a problem.
   - An algorithm shouldn't have computer code. Instead, the algorithm should be written in such a way that, it can be used in similar programming languages

3. **Why every programmer should learn to optimize algorithms?**

   Ans: This article is for those who have just started learning algorithms, and wondered How impactful will it be to boost their career/programming skills. It is also for those who wonder why big companies like Google, Facebook and Amazon hire programmers who are exceptionally good at optimizing Algorithms.

---

## ❖ CSC 247 – Computer Organization & Architecture

- **Structure: The Computer:**

**CPU:**

Controls the operation of the computer and performs its data processing functions – Registers, Arithmetic and Logic Unit, Internal Bus, Control Unit

**Main memory:**

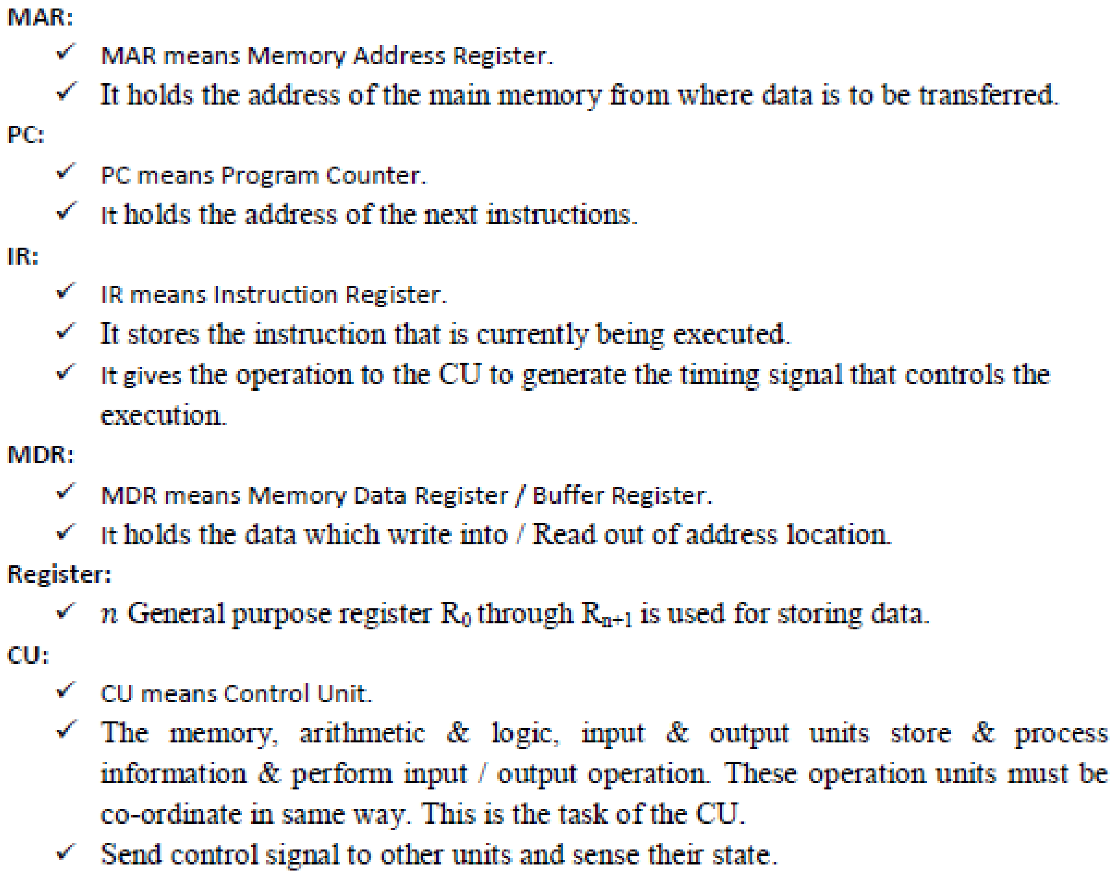
Stores data

**I/O:**

Moves data between the computer and its external environment

**System interconnection:**

Provides for communication among CPU, main memory, and I/O



**MAR:**

- MAR means Memory Address Register.
- It holds the address of the main memory from where data is to be transferred.

**PC:**

- PC means Program Counter.
- It holds the address of the next instructions.

**IR:**

- IR means Instruction Register.
- It stores the instruction that is currently being executed.
- It gives the operation to the CU to generate the timing signal that controls the execution.

**MDR:**

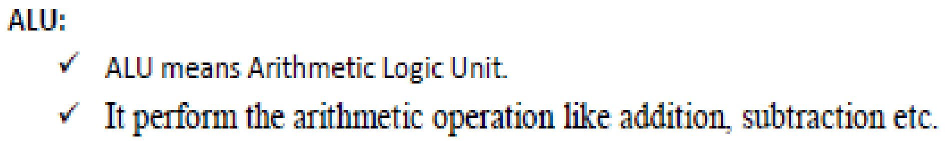
- MDR means Memory Data Register / Buffer Register.
- It holds the data which write into / Read out of address location.

**Register:**

- $n$ General purpose register $R_0$ through $R_{n+1}$ is used for storing data.

**CU:**

- CU means Control Unit.
- The memory, arithmetic & logic, input & output units store & process information & perform input / output operation. These operation units must be co-ordinate in same way. This is the task of the CU.
- Send control signal to other units and sense their state.



**ALU:**

- ALU means Arithmetic Logic Unit.
- It perform the arithmetic operation like addition, subtraction etc.

- **Explain what is Computer Architecture?**

Computer architecture is a specification detailing about How a set of software and hardware standards interacts with each other to form a computer system or platform.

- **How Computer Architecture is characterized?**

The computer architecture is characterized into three categories

- **System Design**: It includes all the hardware component in the system, including data processor aside from the CPU like direct memory access and graphic processing unit
- **Instruction Set Architecture (ISA):** It is the embedded programming language of the central processing unit. It determines the CPU’s functions and capabilities based on programming it can process.
- **Microarchitecture**: It defines the data path, storage element, and data processing as well as How they should be implemented in the ISA.

- **Mention important steps for computer design?**

A CPU architecture is defined by the set of machine language which can be defined as a

- Set of registers and their functions ( capabilities )
- Sequence of micro-operations performed on the data stored in registers
- Control signals that initiate the sequence

- **What are the different types of fields that are part of an instruction?**

The different types of fields that are parts of an instruction are

- Operation Code Field or OP Code field: This field is used to determine the operation to be performed for the instruction
- Address Field: This field is used to determine various addresses such as memory address and register address
- Mode Field: This field determines How operand is to perform or How effective address is derived

- **What are the basic components of a Microprocessor?**

The basic components of a Microprocessor are

- Address lines to refer to the address of a block
- Data lines for data transfer
- IC chips for processing data

- **What are the common components of a microprocessor are?**

The common components of a microprocessor include

- I/O Units
- Control Unit
- Arithmetic Logic Unit (ALU)
- Registers
- Cache
- **SEMICONDUCTOR MAIN MEMORY:** The basic element of a semiconductor memory is the memory cell.
- **RAM:** Random-access memory(RAM)
- **DYNAMIC RAM:** A dynamic RAM (DRAM) is made with cells that store data as charge on capacitors.
- **STATIC RAM:** Static RAM (SRAM) is a digital device that uses the same logic elements used in the processor.
- **ROM:** Read-only memory(ROM) contains a permanent pattern of data that cannot be changed. A ROM is nonvolatile.
- **PROM:** Programmable ROM (PROM) is nonvolatile and may be written into only once.
- **EPROM:** Erasable programmable read-only memory (EPROM) is read and written electrically, as with PROM.
- **EEPROM:** Electrically erasable programmable read-only memory (EEPROM).
- **FLASH memory:** Flash memory is intermediate between EPROM and EEPROM in both cost and functionality.

- **Processor:**

The requirements placed on the processor, the things that it must do:

- **Fetch instruction:** The processor reads an instruction from memory (register, cache, main memory).
- **Interpret instruction:** The instruction is decoded to determine What action is required.

- **Fetch data:** The execution of an instruction may require reading data from memory or an I/O module.
- **Process data:** The execution of an instruction may require performing some arithmetic or logical Operations on data.
- **Write data:** The results of an execution may require writing data to memory or an I/O module.

- **Instruction pipelining:**

✓ It is a process in which several components of the CPU are used to execute more than one instruction concurrently.

## ❖ CSC 433 - Database Management System

- **What is DBMS used for?**

DBMS, commonly known as Database Management System, is an application system whose main purpose revolves around the data. This is a system that allows its users to store the data, define it, retrieve it and update the information about the data inside the database.

- **What is meant by a Database?**

In simple terms, Database is a collection of data in some organized way to facilitate its users to easily access, manage and upload the data.

- **What is the purpose of normalization in DBMS?**

Normalization is the process of analyzing the relational schemas which are based on their respective functional dependencies and the primary keys in order to fulfill certain properties.

**The properties include:**

- To minimize the redundancy of the Data.
- To minimize the Insert, Delete and Update Anomalies.

- **What are the different types of languages that are available in the DBMS?**

Basically, there are 3 types of languages in the DBMS as mentioned below:

- **DDL:** DDL is Data Definition Language which is used to define the database and schema structure by using some set of SQL Queries like CREATE, ALTER, TRUNCATE, DROP and RENAME.

- **DCL:** DCL is Data Control Language which is used to control the access of the users inside the database by using some set of SQL Queries like GRANT and REVOKE.
- **DML:** DML is Data Manipulation Language which is used to do some manipulations in the database like Insertion, Deletion, etc. by using some set of SQL Queries like SELECT, INSERT, DELETE and UPDATE.

- **What is the purpose of SQL?**

SQL stands for **Structured Query Language** whose main purpose is to interact with the relational databases in the form of inserting and updating/modifying the data in the database.

- **Explain the concepts of a Primary key and Foreign Key.**

**Primary Key** is used to **uniquely** identify the records in a database table while **Foreign Key** is mainly used to **link** two or more tables together as this is a particular field(s) in one of the database tables which are the primary key of some other table.

**Example**: There are 2 tables – Employee and Department and both have one common field/column as **'ID'** where ID is the **primary** key of the **Employee** table while this is the **foreign** key for the **Department** table.

- **What are the main differences between Primary key and Unique Key?**

**Given below are few differences:**

- The **main** difference between the Primary key and Unique key is that the Primary key can never have a **null** value while the Unique key may consist of null value.
- In each table, there can be only one primary key while there can be more than one unique key in a table.

- **Explain Entity, Entity Type, and Entity Set in DBMS?**
  - **Entity** is an object, place or thing which has its independent existence in the real world and about which data can be stored in a database.

  - **Entity Type** is a collection of the entities which have the same attributes.

  - **Entity Set** is a collection of the entities of the same type

- **What are the different levels of abstraction in the DBMS?**

There are 3 levels of data abstraction in the DBMS.

**They include:**

- **Physical Level:** This is the lowest level of the data abstraction which states How the data is stored in the database.

- **Logical Level:** This is the next level of the data abstraction which states the type of the data and the relationship among the data that is stored in the database.

- **View Level:** This is the highest level in the data abstraction which sHows/states only a part of the database.

- **What is E-R model in the DBMS?**

E-R model is known as an **Entity-Relationship model** in the DBMS which is based on the concept of the Entities and the relationship that exists among these entities.

- **What is a functional dependency in the DBMS?**

This is basically a constraint which is useful in describing the relationship among the different attributes in a relation.

- **What is 1NF in the DBMS?**

1NF is known as the **First Normal Form**.

This is the easiest form of the normalization process which states that the **domain** of an attribute should have only atomic values. The objective of this is to remove the duplicate columns that are present in the table.

- **What is 2NF in the DBMS?**

2NF is the **Second Normal Form**.

Any table is said to have in the 2NF if it satisfies the following 2 conditions:

- A table is in the 1NF.
- Each non-prime attribute of a table is said to be functionally dependent in totality on the primary key.

- **What is 3NF in the DBMS?**

3NF is the **Third Normal Form.**

Any table is said to have in the 3NF if it satisfies the following 2 conditions:

- A table is in the 2NFm.
- Each non-prime attribute of a table is said to be non-transitively dependent on every key of the table.

- **What is a join in the SQL?**

A Join is one of the SQL statements which is used to join the data or the rows from 2 or more tables on the basis of a common field/column among them.

- **What are different types of joins in the SQL?**

  **There are 4 types of SQL Joins:**

  - **Inner Join:** This type of join is used to fetch the data among the tables which are common in both the tables.
  - **Left Join:** This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.
  - **Right Join:** This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
  - **Full Join:** This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values.

- **What is meant by trigger?**

Trigger is one of the very important codes or programs which get executed **automatically in response to the events** that occur in a table or a view. **Eg:** If a new record is inserted in an employee database then the data gets created automatically in the related tables like salary, department and roles tables.

**What is RDBMS?**

RDBMS is the Relational Database Management System which contains data in the form of the tables and data is accessed on the basis of the common fields among the tables.

- **What are the different type of relationships in the DBMS?**

Relationships in DBMS depicts an association between the tables.

**Different types of relationships are:**

- **One-to-One:** This basically states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables. E**g**: Among a married couple, both wife and husband can have only one spouse.
- **One-to-Many:** This states that there can be many relationships for one i.e. a primary key table hold only one record which can have many, one or none records in the related table. **Eg:** A Mother can have many children.
- **Many-to-Many:** This states that both the tables can be related to many other tables. **Eg:** One can have many siblings and so do they have.

## ❖ CSC 329 – Logic Design & switching circuit

**1. BCD:** Binary Coded Decimal

2. **Convert $(25)_{10}$ to $(?)_{BCD}$**

2=0010, 5=0101

$(25)_{10} = (100101)_{BCD}$

3. **Do 1's complement** of 10010101

= 1's complement = 01101010

4. **Do 2's complement** of 10010101

= 1's complement = 01101010

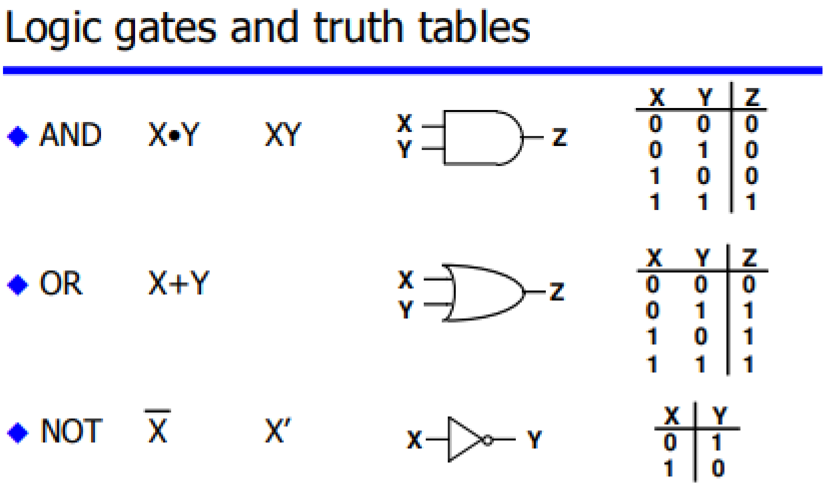
+1

2's complement = 01101011

5. **ASCII:** American Standard Code for Information Interchange

6. **Register**: The device most commonly used for holding data is a register.

7. **Logic Gate**: Logic gates are electronic circuits that operate on one or more input signals to produce an output signal.



Logic gates and truth tables

◆ AND X•Y XY

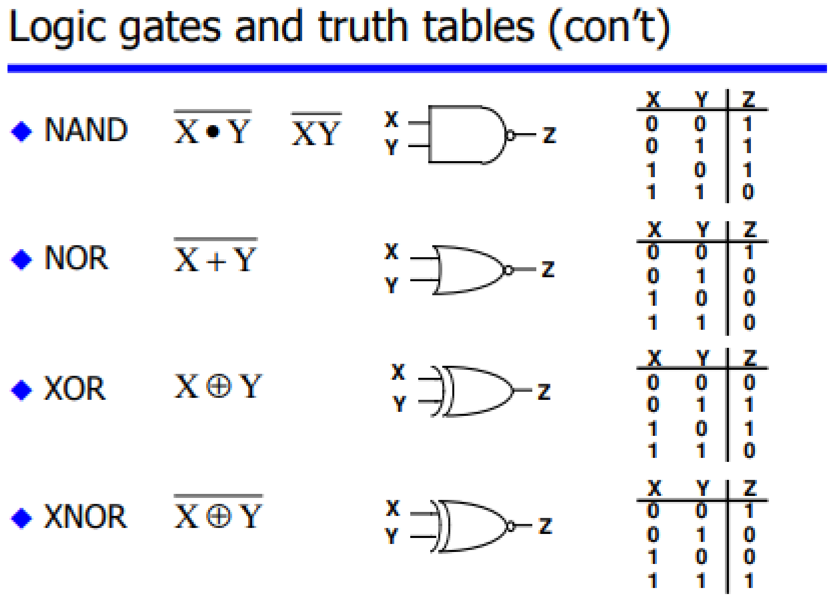
| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

◆ OR X+Y

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

◆ NOT $\overline{X}$ X'

| X | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |



## Logic gates and truth tables (con't)

- NAND $\overline{X \bullet Y}$ $\overline{XY}$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

- NOR $\overline{X + Y}$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

- XOR $X \oplus Y$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

- XNOR $\overline{X \oplus Y}$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

8. **XOR:** Exclusive OR

9. **Binary Adder**: A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers.

10. **Decoder:** A decoder is a combinational circuit that converts binary information from n input lines to a maximum of $2^n$ unique output lines.

11. **Encoder:** An encoder is a combinational circuit that converts binary information from $2^n$ input lines to n output lines.

12. **Multiplexer:** A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs to a single output line.

13. **Memory Unit:** A memory unit is a device to which binary information is transferred for storage from which information is retrieved when needed for processing.

## ❖ CSC 383 – Programming Java

**Q #1) What is JAVA?**

**Ans:** Java is a high-level programming language and is platform independent.

Java is a collection of objects. It was developed by Sun Microsystems. There are a lot of applications, websites and Games that are developed using Java.

**Q #2) What are the features in JAVA?**

**Ans: Features of Java:**

- **Oops concepts**
  - Object-oriented
  - Inheritance
  - Encapsulation
  - Polymorphism
  - Abstraction
- **Platform independent:** A single program works on different platforms without any modification.
- **High Performance:** JIT (Just In Time compiler) enables high performance in Java. JIT converts the bytecode into machine language and then JVM starts the execution.
- **Multi-threaded:** A flow of execution is known as a Thread. JVM creates a thread which is called main thread. The user can create multiple threads by extending the thread class or by implementing Runnable interface.

**Q #3) How does Java enable high performance?**

**Ans:** Java uses Just In Time compiler to enable high performance. JIT is used to convert the instructions into bytecodes.

**Q #4) What are the Java IDE's?**

**Ans:** Eclipse and NetBeans are the IDE's of JAVA.

**Q #5) What do you mean by Constructor?**

**Ans: The points given below explain What a Constructor is in detail:**

- When a new object is created in a program a constructor gets invoked corresponding to the class.

- The constructor is a method which has the same name as class name.
- If a user doesn’t create a constructor implicitly a default constructor will be created.
- The constructor can be overloaded.
- If the user created a constructor with a parameter then he should create another constructor explicitly without a parameter.

**Q #6) What is meant by Local variable and Instance variable?**

**Ans: Local variables** are defined in the method and scope of the variables that have existed inside the method itself.

**An instance variable** is defined inside the class and outside the method and scope of the variables exist throughout the class.

**Q #7) What is a Class?**

**Ans:** All Java codes are defined in a class. A Class has variables and methods.

**Variables** are attributes which define the state of a class.

**Methods** are the place where the exact business logic has to be done. It contains a set of statements (or) instructions to satisfy the particular requirement.

**Example:**

```
public class Addition{ //Class name declaration
int a = 5; //Variable declaration
int b= 5;
public void add(){ //Method declaration
int c = a+b;
}
}
```

**Q #8) What is an Object?**

**Ans:** An instance of a class is called object. The object has state and behavior.

Whenever the JVM reads the “new()” keyword then it will create an instance of that class.

**Example:**

```
public class Addition{
```

```
public static void main(String[] args){
Addion add = new Addition();//Object creation
}
}
```

The above code creates the object for the Addition class.

**Q #9)What are the Oops concepts?**

**Ans: Oops concepts include:**

- Inheritance
- Encapsulation
- Polymorphism
- Abstraction
- Interface

**Q #10) What is Inheritance?**

**Ans:** Inheritance means one class can **extend** to another class. So that the codes can be reused from one class to another class.

Existing class is known as Super class whereas the derived class is known as a sub class.

**Example:**

```
Super class:
public class Manupulation(){
}
Sub class:
public class Addition extends Manipulation(){
}
```

Inheritance is applicable for public and protected members only. Private members can’t be inherited.

**Q #11) What is Encapsulation?**

**Ans: Purpose of Encapsulation:**

- Protects the code from others.
- Code maintainability.

**Example:**

We are declaring ‘a’ as an integer variable and it should not be negative.

```
public class Addition(){
int a=5;
}
```

If someone changes the exact variable as “**a = -5”** then it is bad.

**In order to overcome the problem we need to follow the below steps:**

- We can make the variable as private or protected one.
- Use public accessor methods such as set<property> and get<property>.

**So that the above code can be modified as:**

```
public class Addition(){
private int a = 5; //Here the variable is marked as private
}
```

**Below code sHows the getter and setter.**

Conditions can be provided while setting the variable.

```
get A(){

}

set A(int a){

if(a>0){// Here condition is applied

.........

}

}
```

For encapsulation, we need to make all the instance variables as private and create setter and getter for those variables. Which in turn will force others to call the setters rather than access the data directly.

**Q #12) What is Polymorphism?**

**Ans:** Polymorphism means many forms.

A single object can refer the super class or sub-class depending on the reference type which is called polymorphism.

**Example:**

```
Public class Manipulation(){ //Super class
public void add(){
}
}
public class Addition extends Manipulation(){ // Sub class
public void add(){
}
public static void main(String args[]){
Manipulation addition = new Addition();//Manipulation is reference type and Addition is reference type
addition.add();
}
}
```

Using Manipulation reference type we can call the Addition class “add()” method. This ability is known as Polymorphism.

Polymorphism is applicable for **overriding** and not for **overloading**.

**Q #13) What is meant by Method Overriding?**

**Ans: Method overriding happens if the sub class method satisfies the below conditions with the Super class method:**

- Method name should be same
- Argument should be same
- Return type also should be same

The key benefit of overriding is that the Sub class can provide some specific information about that sub class type than the super class.

**Example:**

```
public class Manipulation{ //Super class

public void add(){

………………

}

}



Public class Addition extends Manipulation(){

Public void add(){

………..

}

Public static void main(String args[]){

Manipulation addition = new Addition(); //Polimorphism is applied

addition.add(); // It calls the Sub class add() method
}

}
```

**addition.add()** method calls the add() method in the Sub class and not the parent class. So it overrides the Super class method and is known as Method Overriding.

**Q #14) What is meant by Overloading?**

**Ans:** Method overloading happens for different classes or within the same class.

**For method overloading, subclass method should satisfy the below conditions with the Super class method (or) methods in the same class itself:**

- Same method name
- Different argument type
- May have different return types

**Example:**

```
public class Manipulation{ //Super class

public void add(String name){ //String parameter

………………

}

}



Public class Addition extends Manipulation(){

Public void add(){//No Parameter

………..

}
Public void add(int a){ //integer parameter

}

Public static void main(String args[]){

Addition addition = new Addition();

addition.add();

}

}
```

Here the add() method having different parameters in the Addition class is overloaded in the same class as well as with the super class.

**Note:** Polymorphism is not applicable for method overloading.

**Q #15) What is meant by Interface?**

**Ans:** Multiple inheritance cannot be achieved in java. To overcome this problem Interface concept is introduced.

An interface is a template which has only method declarations and not the method implementation.

**Example:**

```
Public abstract interface IManupulation{ //Interface declaration
Public abstract void add();//method declaration
public abstract void subtract();
}
```

- All the methods in the interface are internally **public abstract void**.
- All the variables in the interface are internally **public static final** that is constants.

- Classes can implement the interface and not extends.
- The class which implements the interface should provide an implementation for all the methods declared in the interface.

```
public class Manupulation implements IManupulation{ //Manupulation class uses the interface
Public void add(){
……………
}
Public void subtract(){
…………….
}
}
```

**Q #16) What is meant by Abstract class?**

**Ans:** We can create the Abstract class by using “Abstract” keyword before the class name. An abstract class can have both “Abstract” methods and “Non-abstract” methods that are a concrete class.

**Abstract method:**

The method which has only the declaration and not the implementation is called the abstract method and it has the keyword called “abstract”. Declarations are the ends with a semicolon.

**Example:**

```
public abstract class Manupulation{
public abstract void add();//Abstract method declaration
Public void subtract(){
}
}
```

- An abstract class may have a Non- abstract method also.
- The concrete Subclass which extends the Abstract class should provide the implementation for abstract methods.

# ❖ CSC 439 – Visual Programming

- **ASP.NET Core** is a new open-source and cross-platform framework for building modern cloud based internet connected applications, such as web apps, IoT apps and mobile backends. ASP.NET Core apps cross-platform on Windows, Mac and Linux. ASP.NET Core is open source at GitHub

- **The Model-View-Controller (MVC)** architectural pattern separates an app into three main components: Model, View, and Controller. The MVC pattern creates a separation of concerns (SoC) between the model, view, and controller.
  - ✓ The controller's responsibility is to handle the request and to build a model.
  - ✓ The model's responsibility is to transport data and logic between the controller and the view
  - ✓ The view is responsible for transforming that data into HTML.

- **Controller:** The controller is responsible for handling any HTTP requests that come to the application. The controller's responsibility is then to gather and combine all the necessary data and package it in model objects, which act as data carriers to the views.

- **Models:** Classes that represent the data of the app. The model classes use validation logic to enforce business rules for that data. Typically, model objects retrieve and store model state in a database.

- **View:** Views are the components that display the app's user interface (UI). Generally, this UI displays the model data.

- **MVC Logic:** MVC pattern specifies where each kind of logic should be located in the application. Logics are-
  - ✓ Input Logic: Input logic belongs in the Controller.
  - ✓ Business Logic: Business logic belongs in the model.
  - ✓ UI Logic: UI logic belongs in  the View.

This separation helps us to reduce complexity when we build an application.

- **Routing:** ASP.NET Core MVC uses the Routing middleware to match the URLs of incoming requests and map them to actions.
  Routing is two types- Convention bsed routing, Attribute routing

- **Identifiers and Keywords:**

  **Identifiers** are names that programmers choose for their classes, methods, variables, and so on.

  Example-

  System Test Main x Console WriteLine

  **Keywords** are names that mean something special to the compiler.

  Example-

  using class static void int

- **Type:**

  A **type** defines the blueprint for a value.

  A **variable** denotes a storage location that can contain different values over time.

  A c**onstant** always represents the same value.

  Example: const int y = 360;

- **Predefined Type:** Predefined types are types that are specially supported by the compiler. The int type is a predefined type for representing the set of integers that fit into 32 bits of memory, from −231 to 231−1, and is the default type for numeric literals within this range.

- **The predefined types in C# are:**

  **1. Value types**

  **Numeric**

  Signed integer (sbyte, short, int, long)

  Unsigned integer (byte, ushort, uint, ulong)

  Real number (float, double, decimal)

**Logical (bool)**

**Character (char)**

**2. Reference types**

**String (string)**

**Object (object)**

- **Reference Type** variables are stored in the heap while **Value Type** variables are stored in the stack.

- **The Console class** is actually a static class, which means all its members are static. You never actually create instances of a Console—one console is shared across the whole application.

- **Conversions:** C# can convert between instances of compatible types. A conversion always creates new value from an existing one. Conversions can be either implicit or explicit:

  **Implicit conversions** happen automatically'

  **Explicit conversions** require a cast.

- **Loop:** A loop statement allows us to execute a statement or group of statements multiple times.

- **Various Looping Statement:**

  **for loop** Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

  **while loop** Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body

  **do...while loop** It is more like a while statement, except that it tests the condition at the end of the loop body

  **Foreach loop** it is used to cycle through the elements of a collection

- **Enumuration:** An enumeration is a set of named integer constants. The keyword **enum** declares an enumerated type. The general form for an enumeration is

  **enum name { enumeration list };**

- **Razor:** Razor is a mark up syntax that lets us embed server-based code (for example- C#) into web pages. Razor view template file help us to generate the Html response.

  The Razor syntax consists of Razor markup, C#, and HTML.

- **Rules of Razor:**
  - ✓ Razor code blocks are enclosed in @{ ... }
  - ✓ Inline expressions (variables and functions) start with @
  - ✓ Code statements end with semicolon
  - ✓ Strings are enclosed with quotation marks
  - ✓ C# code is case sensitive
  - ✓ C# files have the extension .cshtml

- **@model Directive:** The @model directive specifies the type of the model passed to a view.

- **Interface:** An interface looks like a class but has no implementation.

  The only thing it contains is declarations of events, indexers, methods and/or properties. Interface members are always implicitly public and cannot declare an access modifier.

- **Reason to use of Interface:**

  The reason interfaces only provide declarations is because they are inherited by classes and structs, which must provide an implementation for each interface member declared.

- **Entity Framework (EF)** is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects.

- **Constructor:** A constructor is a special member function of a class that is executed whenever we create new objects of that class.

- **Inheritance:** Inheritance is one of the primary concepts of object-oriented programming.

  A class can inherit from another class to extend or customize the original class.

  A class can inherit from only a single class, but can itself be inherited by many classes.

  Benefit:

  - ✓ Inheriting from a class lets us reuse the functionality in that class instead of building it from scratch.

- **Sealed Class:** Sealed classes are used to restrict the users from inheriting the class.

  A class can be sealed by using the sealed keyword. The keyword tells the compiler that the class is sealed, and therefore, cannot be extended. No class can be derived from a sealed class. Sealed method is implemented so that no other class can overthrow it and implement its own method

- **MVC -LINQ:**

  LINQ (Language-Integrated Query) is used to write LINQ queries in C# for databases, XML documents, ADO.NET Datasets and any collection of objects. LINQ provides you three different ways to write a LINQ query in C#-

  1. Query Expression (Query Syntax)
  2. Method Invocation (Method Syntax)
  3. Mixed Syntax

- **LINQ – Mixed syntax:** You can use a mixture of both syntax by enclosing a query expression inside parentheses and make a call to method.

- **LINQ query operations consist of five parts.**

  - ✓ Obtaining a Data Source
  - ✓ Filtering
  - ✓ Ordering
  - ✓ Joining
  - ✓ Grouping

## ❖ CSC 471 - Microprocessor

1. **What is a Microprocessor?**

   Answer: Microprocessor is a program-controlled device, which fetches the instructions from memory, decodes and executes the instructions. Most Micro Processor are single- chip devices.

2. **What are the various registers in 8085?**

   Answer: Accumulator register, Temporary register, Instruction register, Stack Pointer, Program Counter are the various registers in 8085

3. **What are the types of buses?**

   Answer: There are three types of buses.

   Address bus: This is used to carry the Address to the memory to fetch either Instruction or Data.

   Data bus : This is used to carry the Data from the memory.

   Control bus : This is used to carry the Control signals like RD/WR, Select etc.

4. **What does EU do?**

   Answer: Execution Unit receives program instruction codes and data from BIU, executes these instructions and store the result in general registers.

5. **What are the various segment registers in 8086?**

   Answer: Code, Data, Stack, Extra Segment registers in 8086.

6. **What is Program counter?**

   Answer: Program counter holds the address of either the first byte of the next instruction to be fetched for execution or the address of the next byte of a multi byte instruction, which has not been completely fetched. In both the cases it gets incremented automatically one by one as the instruction bytes get fetched. Also Program register keeps the address of the next instruction.

7. **Which interrupt has the highest priority?**

   Answer: TRAP has the highest priority

8. **What is Stack Pointer?**

   Answer: Stack pointer is a special purpose 16-bit register in the Microprocessor, which holds the address of the top of the stack.

9. **How many interrupts are there in 8085?**

   Answer: There are 12 interrupts in 8085.

10. **What is the function of accumulator?**

    Answer: Accumulator is an 8 bit register which stores data and performs arithmetic and logical operations. The result of the operation is stored in the accumulator. It is designated by the letter ‘A’.

11. **What does CU do?**

    Answer: The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operation of the processor. It tells the computer's memory, arithmetic/logic unit and input and output devices How to respond to the instructions that have been sent to the processor.

12. **What does ALU do?**

    Answer: An arithmetic logic unit (ALU) is a digital circuit used to perform arithmetic and logic operations. It represents the fundamental building block of the central processing unit (CPU) of a computer. Modern CPUs contain very powerful and complex ALUs.

# ❖ CSC 387 – System Analysis & Design

- **System analysis:** System analysis is a method of figuring out the basic elements of a project and deciding How to combine them in a best way to solve a problem.

Three primary roles of system analyst:

- Consultant
- Supporting expert
- Agent of change

- **SDLC:** It stands for Systems Development Life Cycle which is a phased approach to solving business problems.

**The Seven Phases of the Systems Development Life Cycle:**

1. Identifying Problems, Opportunities, and Objectives
2. Determining Human Information Requirements
3. Analyzing System Needs
4. Designing the Recommended System
5. Developing and Documenting Software
6. Testing and Maintaining the System
7. Implementing and Evaluating the System

**CASE tools:** CASE tools are productivity tools for systems analysts that have been created explicitly to improve their routine work through the use of automated support.

Reasons for using CASE tools

- Increasing analyst productivity
- Improving analyst-user communication
- Integrating life cycle activities

**ERP:** ERP stands for Enterprise Resource Planning described an integrated organizational information system.

**Agile Approach:** The Agile Method is a particular approach to project management that is utilized in software development.

Five Stages of Agile Development:

- Exploration
- Planning
- Iterations to the first release
- Productionizing
- Maintenance

- **Context-Level Data Flow Diagrams**: It focus is on the data flowing into and out of the system and the processing of the data. The Basic Symbols of a Data Flow Diagram:
    1. Process: A process means some actions take place
    2. Entity: An Entity is a person, group, or any system that receives or originates information or data.
    3. Data flow- It sHows information being is passed from or to a process.

- **Entity-Relationship Model:** An entity-relationship diagram (**ERD**) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities.
    1. Entity: An Entity is a person, group, or any system that receives or originates information or data.
    2. Attribute: An attribute is a characteristic of an entity.

- **Use case diagram:** A use case diagram is a graphic depiction of the interactions among the elements of a system.

**Actor:** An actor is a user of the system playing a particular role.
**Use case:** Use case is a particular activity a user can do on the system.
**Relationship:** Relationships are simply illustrated with a line connecting actors to use cases.

- **Feasibility Study**

Feasibility study determines whether that solution is feasible or achievable for the organization.

1. Technical Feasibility: Technical feasibility addresses concerns about hardware capability, reliability and availability and the skills of the development team.
2. Economic feasibility: Economic feasibility determines to What extent a new system, is cost effective.
3. Operational feasibility: Operational feasibility addresses concerns about user acceptance, management support, and then requirements of entities and factors in the organizations external environment.

- **BYOD:** Bring your own device
- **BYOT:** Bring your own technology

- **Tangible benefit:** Tangible benefits are those measured in monetary terms. Examples:
    - Increase in the speed of processing
    - The advantage of the computer's superior calculating power
    - Decreases in the amount of employee time needed to complete specific tasks

- **Intangible benefits:** Intangible benefits cannot be measured in monetary terms but they do have a very significant business impact. Examples:
    - Becoming more competitive in customer service
    - Increasing job satisfaction

- **Break-Even Analysis:** Break even analysis is the point at which the total cost of the current system and the proposed system intersect.

- **Work Breakdown Structure:** Often a project needs to be broken down into smaller tasks or activities. These tasks together make up a work breakdown structure (WBS).

- **Function Point Analysis:** Function Point Analysis (FPA) refers to the practice of using function points to size and estimate the cost of work on systems

Takes the five main components of a computer system and rates them in terms of complexity:

- External inputs
- External outputs
- External queries
- Internal logical files
- External interface files

- **Interviewing:** Interviewing is an important method for collecting data on human and system information requirements.

**Question Types:**

1. Open ended: Open-ended interview questions allow interviewees to respond How they wish, and to What length they wish
2. Closed: Closed interview questions limit the number of possible responses

**Bipolar Questions:** Bipolar questions are those that may be answered with a “yes” or “no” or “agree” or “disagree”

**Arranging Questions:**

- Pyramid -Starting with closed questions and working toward open-ended questions
- Funnel - Starting with open-ended questions and working toward closed questions
- Diamond - Starting with closed, moving toward open-ended, and ending with closed questions

**Sampling:** A process of systematically selecting representative elements of a population

**Investigation:** The act of discovery and analysis of data

- **Software prototyping:** It is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed.

**Types of software prototyping-**

- Patched-up : A working model that has all the features but is inefficient
- Nonoperational: A nonworking scale mode that is set up to test certain aspects of the design
- First-of-a-series: A full-scale prototype is installed in one or two locations first, and if successful, duplicates are installed at all locations based on customer usage patterns and other key factors.
- Selected features: Building an operational model that includes some, but not all, of the features that the final system will have.

- **Agile Modeling:** Agile methods are a collection of innovative, user-centered approaches to systems development.

**Four Basic Activities of Agile Modeling:**

- Coding, Testing, Listening, Designing

**Four Resource Control Variables of Agile Modeling**

- Time, Cost, Quality, Scope

**Aliases:** Synonyms or other names for the element

- **Logical and Physical Data Structure:**
    - **Logical:** SHow What data the business needs for its day-to-day operations
    - **Physical:** Include additional elements necessary for implementing the system

- **Objects:** Persons, places, or things that is relevant to the system being analyzed
- **Classes:** Defines the set of shared attributes and behaviors found in each object
- **Inheritance:** When a derived class inherits all the attributes and behaviors of the base class
- **Cipher code**: A cipher (or cypher) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure.

- **Quality of a system analyst:**
    1. Good learning and listening skills
    2. Complex problem solving ability.
    3. Time management
    4. Critical thinking
    5. Good communication skills.

# ❖ CSC 397 – Theory of Computation

1. **What is set?**

   Answer: A set is a group of objects as a unit

2. **What is element?**

   Answer: The objects present in a set are elements.

3. **What is proposition?**

   Answer: Proposition is a declarative statement (declares a fact) that have a truth value (true or false)

4. **What is function?**

   Answer: Function is an object that sets up an input-output relationship.

5. **What is alphabet?**

   Answer: A finite set of symbols that can be used to form strings in the language.

6. **What is string?**

   Answer: A string over an alphabet is a finite ordered sequence of symbols from the set of alphabet.

7. **What is language?**

   Answer: A language L (G) defined by the grammar G is the set of all sentences derivable using grammar G

8. **What is grammar?**

   Answer: It is a system that specifies How the characters of an alphabet S can be joined (concatenate) to form a legal string in a language.

9. **What is algorithm?**

   Answer: An algorithm is detailed and unambiguous sequence of instructions that produces a result.

**10. What is flowchart?**

Answer: Flowchart is the visual representation of algorithm drawn with the help of basic geometric figures.

**11. What is state diagram?**

Answer: If the figures of the flow chart are replaced by circles, a directed graph can be obtained which consists of a sequence of internal configuration of system called states. This diagram is call state diagram.

**12. What is finite automata?**

Answer: Finite automaton is the simplest model for computers with an extremely limited amount of memory. Finite automaton is used to design algorithms.

**13. How many types of finite automata are there?**

Answer: There are two types:

I. Deterministic finite automata DFA
II. Non-deterministic finite automata NFA

**14. What is non-deterministic finite automata?**

Answer: In an NFA the transition function takes a state and an input symbol or the empty string ε and produces the set of possible next states.

**15. What is regular expression?**

Answer: Expressions build up using the regular operations for describing languages.

**16. What is generalized nondeterministic finite automata?**

Answer: GNFA are simply nondeterministic finite automata wherein the transition arrows may have any regular expressions as labels.

**17. What is context-free grammars?**

Answer: Context-free grammars describes features that have a recursive structure.

**18. What is derivation?**

Answer: The sequence of substitutions to obtain a string is called a derivation.

**19. What is leftmost derivation?**

Answer: A derivation of a string w in a grammar G is a **leftmost** derivation if at every step the leftmost remaining variable is the one replaced.

**20. What is ambiguous grammar?**

Answer: A grammar is ambiguous if it generates the same string in several different ways.

**21. What is pushdown automata?**

Answer: Pushdown automata is a class of machines recognizing the context-free languages.

**22. What is Turing machine?**

Answer: A much more powerful and accurate model of a general purpose computer model, Similar to a finite automaton but with an unlimited and unrestricted memory.

**23. What is Enumerator?**

Answer: It is a Turing machine with an attached printer.

**24. What is asymptotic analysis?**

Answer: Asymptotic analysis is a convenient form of estimation of running time, of the algorithm when it is run on large inputs.

**25. What is big-O notation?**

Answer: Big-O notation says that one function is asymptotically no more than another.

**26. What is small-o notation?**

Answer: Small-o notation is used to say that one function is asymptotically less than another.

**27. What is path?**

Answer: A potential path is a sequence of nodes in a directed graph G having a length of at most m, where m is the number of nodes in G.

## ❖ CSC 465 – Data Communication & Networks

**1. What is data communication?**

**Ans:** Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable.

The effectiveness of a data communications system depends on four fundamental characteristics: **delivery (**The system must deliver data to the correct destination.), **accuracy** (The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable), **timeliness** (The system must deliver data in a timely manner), and **jitter** (Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets.).

**2. What are the data communication component?**

**Ans:**

I. Message (The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.)

II. Sender (The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.)

III. Receiver (The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on).

IV. Transmission medium (The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.)

V. Protocol (A protocol is a set of rules that govern data communications.)

**3. What are the data flow?**

**Ans:** Communication between two devices can be **simplex** (the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive. Like keyboard and monitor), **half-duplex** (In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa. Like Walkie-talkies and CB (citizens band) radio), **full-duplex** (In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously. The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. Like telephone).

**4. What is networking?**

**Ans:** A network is a set of devices (often referred to as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network. A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

**5. What are the network topology?**

Ans: There are four basic topologies possible: mesh, star, bus, and ring.

**6. What are the Categories of Networks?**

**Ans:** 1. Personal Area Network (PAN)

The smallest and most basic type of network, a PAN is made up of a wireless modem, a computer or two, phones, printers, tablets, etc., and revolves around one person in one building. These types of networks are typically found in small offices or residences, and are managed by one person or organization from a single device.

2. Local Area Network (LAN)

We’re confident that you’ve heard of these types of networks before – LANs are the most frequently discussed networks, one of the most common, one of the most original and one of the simplest types of networks. LANs connect groups of computers and low-voltage devices together across short distances (within a building or between a group of two or three buildings in close proximity to each other) to share information and resources. Enterprises typically manage and maintain LANs.

3. Wireless Local Area Network (WLAN)

Functioning like a LAN, WLANs make use of wireless network technology, such as Wi Fi. Typically seen in the same types of applications as LANs, these types of networks don’t require that devices rely on physical cables to connect to the network.

4. Campus Area Network (CAN)

Larger than LANs, but smaller than metropolitan area networks (MANs, explained below), these types of networks are typically seen in universities, large K-12 school districts or small businesses. They can be spread across several buildings that are fairly close to each other so users can share resources.

5. Metropolitan Area Network (MAN)

These types of networks are larger than LANs but smaller than WANs – and incorporate elements from both types of networks. MANs span an entire geographic area (typically a town or city, but sometimes a campus). Ownership and maintenance is handled by either a single person or company (a local council, a large company, etc.).

6. Wide Area Network (WAN)

Slightly more complex than a LAN, a WAN connects computers together across longer physical distances. This allows computers and low-voltage devices to be remotely connected to each other over one large network to communicate even when they’re miles apart.

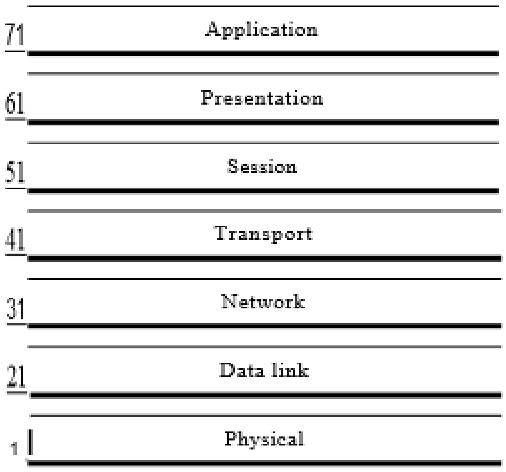
7. Virtual Private Network (VPN)

By extending a private network across the Internet, a VPN lets its users send and receive data as if their devices were connected to the private network – even if they're not. Through a virtual point-to-point connection, users can access a private network remotely.

**7. What is OIS model and layer?**

**Ans:** e International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection model. ISO is the organization. OSI is the model.

Figure 2.2 *Seven layers of the OSI model*



I. The physical layer is responsible for movements of individual bits from one hop (node) to the next

II. The datalink layer is responsible for moving frames from one hop (node) to the next.

III. The network layer is responsible for the delivery of individual packets from the source host to the destination host.

IV. The transport layer is responsible for the delivery of a message from one process to another.

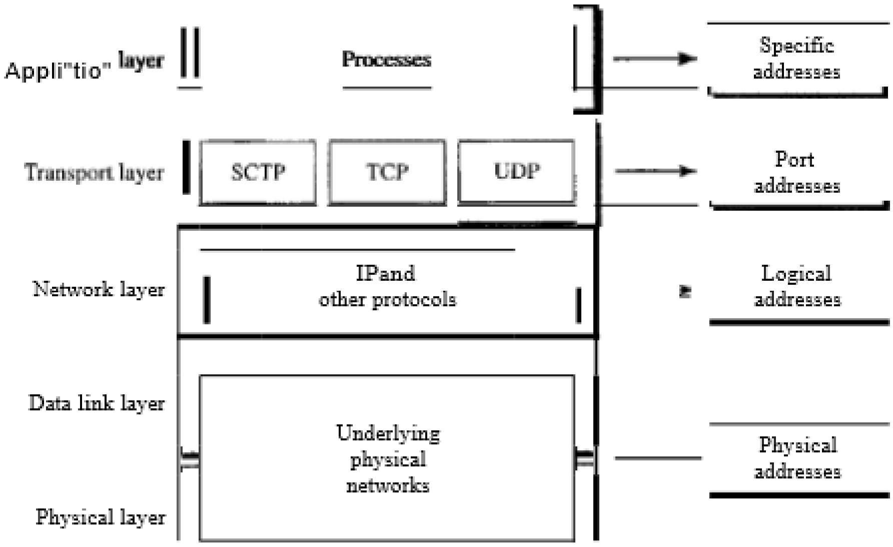
V. The session layer is responsible for dialog control and synchronization.

VI. The presentation layer is responsible for translation, compression, and encryption.

VII. The application layer is responsible for providing services to the user.

**8. What is IP and Layer?**

**Ans:** The TCPIIP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application.



**9. What is Port and spacific Address?**

**Ans:**

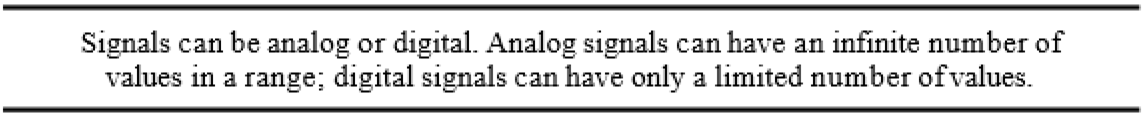
**Port:**The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete.

**Specific Addresses:**

Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, forouzan@fhda.edu) and the Universal Resource Locator (URL).

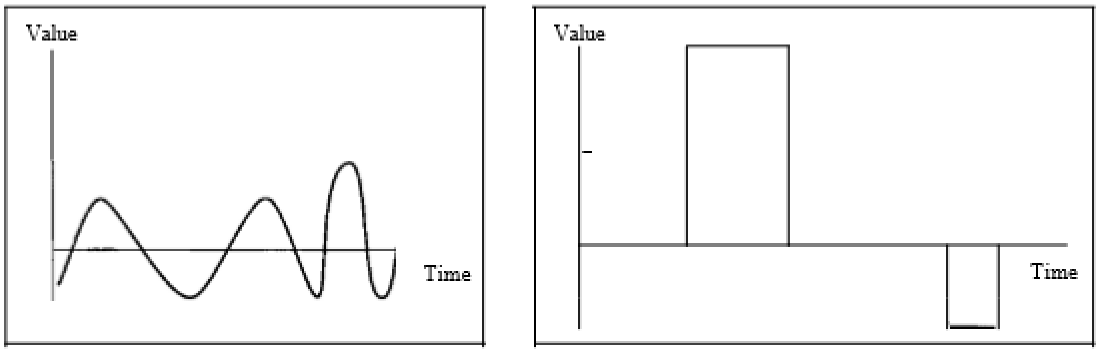
**10. What is analog and digital?**

**Ans:**



Signals can be analog or digital. Analog signals can have an infinite number of values in a range; digital signals can have only a limited number of values.

Figure 3.1 *Comparison ofanalog and digital signals*



a. Analog signal

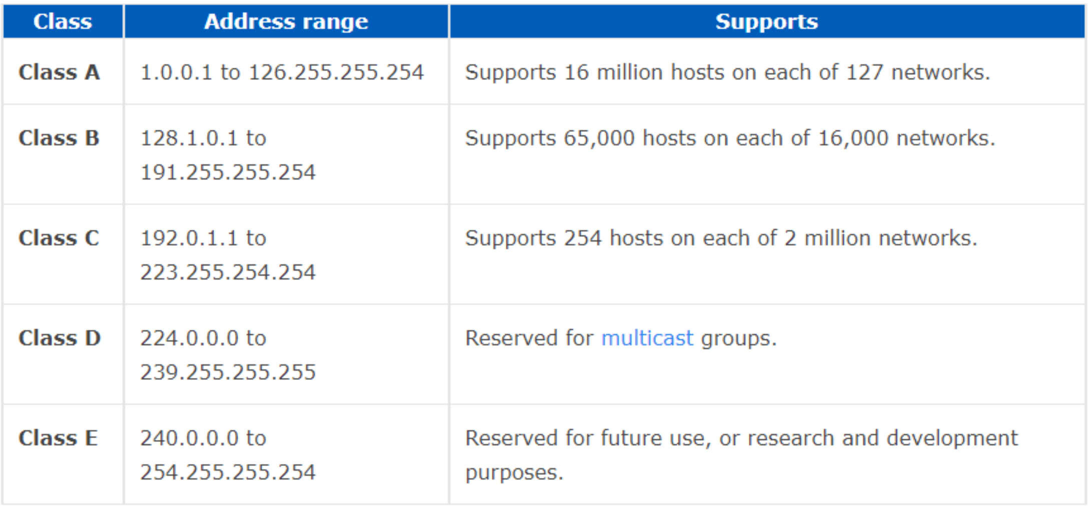
b. Digital signal

**11. What is IP address classes?**

**Ans:**

IPv4 addresses are 32 bits long (four bytes). An example of an IPv4 address is **216.58.216.164**

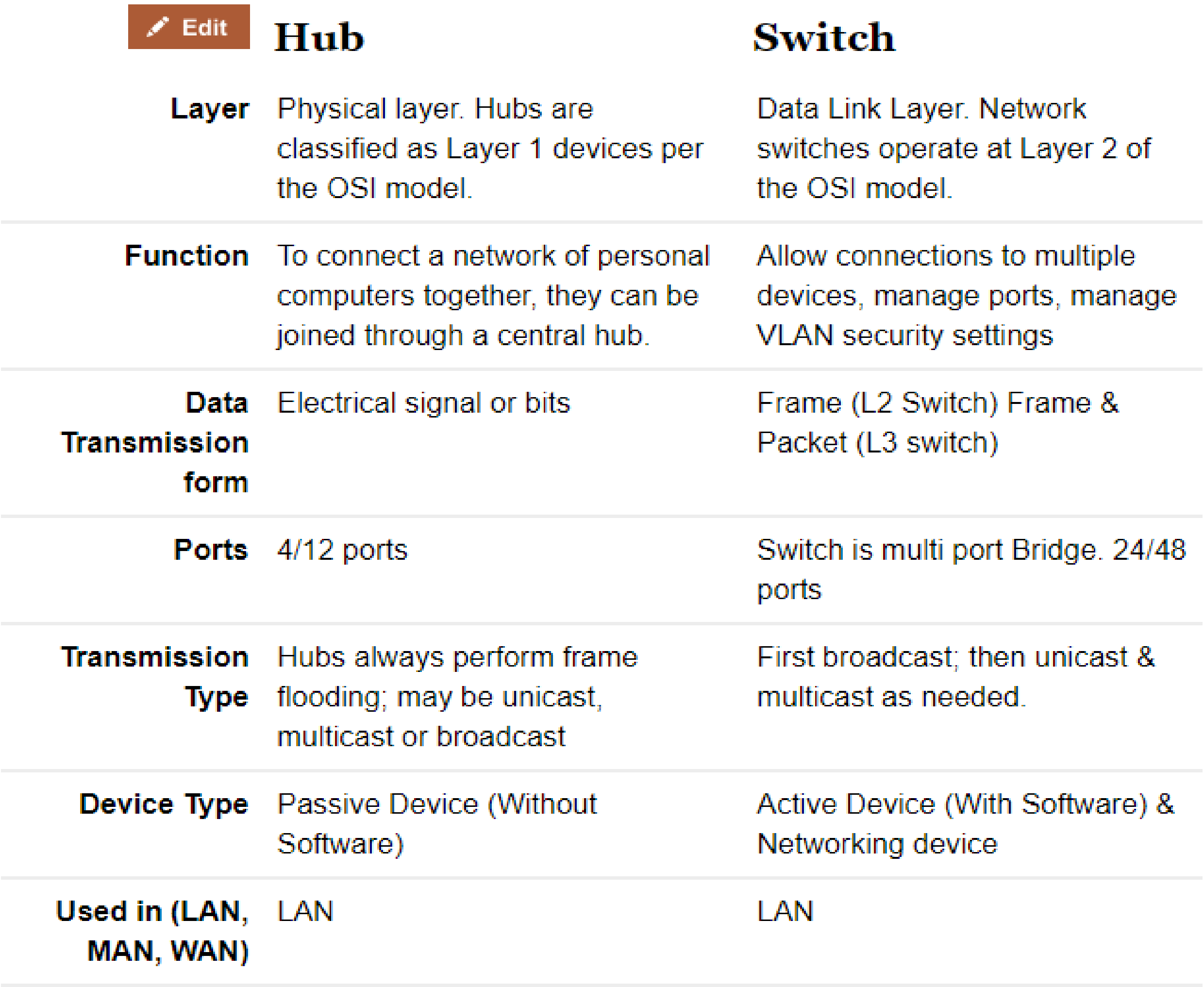
With an IPv4 IP address, there are five classes of available IP ranges: Class A, Class B, Class C, Class D and Class E, while only A, B, and C are commonly used. Each class allows for a range of valid IP addresses, sHown in the following table.



| Class | Address range | Supports |
|---|---|---|
| Class A | 1.0.0.1 to 126.255.255.254 | Supports 16 million hosts on each of 127 networks. |
| Class B | 128.1.0.1 to 191.255.255.254 | Supports 65,000 hosts on each of 16,000 networks. |
| Class C | 192.0.1.1 to 223.255.254.254 | Supports 254 hosts on each of 2 million networks. |
| Class D | 224.0.0.0 to 239.255.255.255 | Reserved for multicast groups. |
| Class E | 240.0.0.0 to 254.255.255.254 | Reserved for future use, or research and development purposes. |

**12. What is differet Hub and switch?**

**Ans:**



| Edit | Hub | Switch |
|---|---|---|
| **Layer** | Physical layer. Hubs are classified as Layer 1 devices per the OSI model. | Data Link Layer. Network switches operate at Layer 2 of the OSI model. |
| **Function** | To connect a network of personal computers together, they can be joined through a central hub. | Allow connections to multiple devices, manage ports, manage VLAN security settings |
| **Data Transmission form** | Electrical signal or bits | Frame (L2 Switch) Frame & Packet (L3 switch) |
| **Ports** | 4/12 ports | Switch is multi port Bridge. 24/48 ports |
| **Transmission Type** | Hubs always perform frame flooding; may be unicast, multicast or broadcast | First broadcast; then unicast & multicast as needed. |
| **Device Type** | Passive Device (Without Software) | Active Device (With Software) & Networking device |
| **Used in (LAN, MAN, WAN)** | LAN | LAN |

| | | |
|---|---|---|
| **Table** | A network hub cannot learn or store MAC address. | Switches use content accessible memory CAM table which is typically accessed by ASIC (Application Specific integrated chips). |
| **Transmission Mode** | Half duplex | Half/Full duplex |
| **Broadcast Domain** | Hub has one Broadcast Domain. | Switch has one broadcast domain [unless VLAN implemented] |
| **Definition** | An electronic device that connects many network device together so that devices can exchange data | A network switch is a computer networking device that is used to connect many devices together on a computer network. A switch is considered more advanced than a hub because a switch will on send msg to device that needs or request it |
| **Speed** | 10Mbps | 10/100 Mbps, 1 Gbps |

# ❖ CSC 451 – Management Information System

## 1. What is MIS?

A Management Information System is an information system that evaluates, analyzes, and processes an organization's data to produce meaningful and useful information based on which the management can take right decisions to ensure future growth of the organization.

## 2. What is Management?

Management refers to a set of functions and processes designed to initiate and coordinate group efforts in an organized setting directed towards promotion of certain interests, preserving certain values and pursuing certain goals.

## 3. What is Information?

Information is data that have been organized into a meaningful and useful context.

## 4. What is System?

**System**: A set of components that work together to achieve a common goal.

**Subsystem**: One part of a system where the products of more than one system are combined to reach an ultimate goal

**Closed system**: Stand-alone system that has no contact with other systems

**Open system**: System that interfaces with other systems

**Characteristics of Information:**

- Timeliness
- Purpose
- Mode and Format
- Redundancy
- Rate
- Frequency
- Completeness

- Reliability
- Cost benefit analysis
- Validity
- Quality

5. **Five characteristics of a good Management Information System:**
   - Management Oriented
   - Management directed
   - Integrated
   - Common data flows
   - Heavy planning element

6. **What are the three approaches of MIS development?**
   - Top down approach
   - Bottom-up approach
   - Integrative approach

7. **What are the three levels of management?**
   - Strategic level
   - Tactical decisions
   - Supervisory level

8. **Limitations of the Management Information System:**
   - MIS is not a substitute for effective management. It cannot replace managerial Judgement in making decisions in different functional areas.
   - MIS may not have requisite flexibility to quickly update itself with changing needs of time.
   - MIS cannot provide tailor-made information packages suitable for the purpose of every type of decision made by executives.
   - MIS takes into account mainly quantitative factors, thus it ignores the non-quantitative factors like morale and attitude of members of the organization.
   - MIS is less useful for making non-programmed decisions.
   - The effectiveness of MIS decreases due to frequent changes in top managements, organizational structure and operational team.

- MIS effectiveness is reduced where culture of hoarding information and not sharing with others exists.

### 9. What is Data Mining?

Data Mining is the process of collecting large amounts of raw data and transforming that data into useful information.

**Advantages**

- Improves Customer Satisfaction/service
- Saves Time and Money
- Increases Sales Effectiveness
- Increases profitability

**Disadvantages**

- Require skilled technical users to interpret and analyze data from warehouse
- Validity of the patterns
- Related to real world circumstances
- Unable to Identify Casual Relationships
- Reserved for the few instead of the many

**Data Mining Applications**

- Banking
- Insurance
- Medicine/Healthcare
- Retail

### 10. Data Warehousing?

A Data Warehouse is a computerized collection of mined data.

**Advantages**

- Access to information
- Data Inconsistency
- Decrease Computing Cost

- Productivity Increase
- Increase company profits

**Disadvantages**

- Data must be cleaned, loaded, and extracted
- User Variability
- Difficult to Maintain

## 11. What is Data Mart?

A data mart is a simple form of a data warehouse that is focused on a single subject (or functional area), such as Sales, Finance, or Marketing.

# ❖ CSC 347 – Computer Hardware & Maintenance

**1. What Is A Computer?**

Computer is a programmable machine. It the integral part of everyday life.

**2. What Are The Different Functions Of A Computer?**

A computer does the following functions;

a) Accepting data

b) Processing Data

c) Storing Data

d) Displaying Data

**3. How A Minicomputer Different From A Mainframe?**

Minicomputer is a midsized multiprocessing and multi user computer. It is also called mid-range server. But mainframes are huge computers, most commonly occupying entire rooms or floor. It is highly costly.

**4. What Is Super Computer?**

The fastest type of computer. Supercomputers are very expensive and are employed for specialized applications that require immense amounts of mathematical calculations. For example, weather forecasting requires a supercomputer. Other uses of supercomputers include animated graphics, fluid dynamic calculations, nuclear energy research, and petroleum exploration.

**5. Differentiate Input And Output Device?**

Input devices are used for giving input to the computer. But output devices are used to get the result back from the computer. The examples of input devices are keyboard, mouse, scanner, digital camera atc...whereas output devices include monitor, printer, projector etc....

**6. What Is A Storage Device? What Is The Common Classification?**

Storage devices are used to store data in the computer. The different types of storage devices are:

a) Magnetic Devices.

b) Optical Devices.

c) Solid-State Storage Devices.

**7. What Do You Mean By A Processing Device? What Are The Various Types Of Processing Devices?**

The main function of a computer is to process data. The various types of processing device in a computer are:

a) Microprocessor

b) Chipset

c) BIOS

**8. Differentiates Serial And Parallel Port?**

Serial port and parallel port are used for transferring data in/out of the computer. In serial port transmission only 1 bit is transmitted at a time. Most serial ports on personal computers conform to the RS-232C or RS-422 standards. A parallel interface for connecting an external device such as a printer. On PCs, the parallel port uses a 25-pin connector (type DB-25) and is used to connect printers, computers and other devices that need relatively high bandwidth. It uses parallel transmission of data.

**9. What Is An Interface?**

These are the communication channel that enables your computer to exchange information with various devices.

**10. What Are The Differences Between Multitasking And Multiprocessing?**

**Multitasking:** Enables the processor to do multiple programs simultaneously by fast switching through the programs. Here doesn't have the involvement of multiple processors.

**Multiprocessing:** Enables the processor to do multiple programs simultaneously by the use of multiple processors.

**11. What Is Cisc And Risc?**

Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC) are two philosophies by which computer chips are designed. RISC became a popular technology buzzword in the 1990s, and many processors used in the enterprise business segment were RISC-based.

**12. What Is Cache Memory? What Is The Advantage If A Processor With More Cache Memory You Are Using?**

Cache memory is the memory area between RAM and Processor. If cache memory increases the speed of the system will also improve.

**13. What Are The Different Types Of Ram?**

SRAM, DRAM, VRAM, SGRAM, DDR-SDRAM etc….

**14. Differentiate Sram And Dram?**

**SRAM**

Static RAM stores each bit of data on six metal oxide semiconductor field effect transistors, or MOSFETs. SRAM is used in devices that require the fastest possible data access without requiring a high capacity. Some examples are CPU caches and buses, hard drive and router buffers and printers.

**DRAM**

Dynamic RAM stores data using a paired transistor and capacitor for each bit of data. Capacitors constantly leak electricity, which requires the memory controller to refresh the DRAM several times a second to maintain the data.

**15. What Are The Different Dram Types?**

FPMDRAM, EDO DRAM, SDRAM, RDRAM, DDR-SDRAM

**16. How does a RAM function?**

Random Access Memory, or RAM, is used by computers to store and access information using a random order. This resource provides a temporary way for computers to process electronic data, while making computers more responsive and helping operate memory-intensive programs. RAM chips are only capable of storing and accessing information when electrical power is present.

**17.  What are differences between RAM and ROM?**

The main difference between RAM and ROM is Ram is volatile memory and  ROM is non-volatile memory.

Other differences between a ROM and a RAM chip include:

- A ROM chip is used primarily in the startup process of a computer, whereas a RAM chip is used in the normal operations of a computer after starting up and loading the operating system.
- Writing data to a ROM chip is a slow process, whereas writing data to a RAM chip is a faster process.
- A RAM chip can store multiple gigabytes (GB) of data, up to 16 GB or more per chip; A ROM chip typically stores only several megabytes (MB) of data, up to 4 MB or more per chip.

**18. What is a flash memory?**

**Flash memory** is an electronic non-volatile computer storage medium that can be electrically erased and reprogrammed. Flash memory is popular in modemsbecause it enables the modem manufacturer to support new protocols as they become standardized. Flash memory is also called *flash RAM.*

**19. What is the purpose of RAM in a computer?**

Ans: RAM (random acces memory) Is basically the memory that runs programs, it doesn't store any long term data unlike a hard drive disk, but its more instant, the less RAM you have the slower your operating system will run, because it doesn't have the room to move everything about.

**20. What kind of memory is a RAM categorized as?**

Ans: RAM categorized into following types.

- DRAM
- SRAM
- DRDRAM

**21. How does the CPU and the RAM communicate?**

Ans: The CPU and the Memory are connected to the north bridge the cpu communicates through the Front Side Bus and the memory is connected via the memory bus.

**22. What is bursting?**

The process of increasing data throughput while RAM transmitting data repeatedly.

**23. What is buffered RAM?**

Ans: Buffered memory is a type of computer memory. It is designed to control the amount of electrical current which goes to and from the memory chips at any one time. This makes for more stable memory, but increases the cost and slows the speed at which it works. In a buffered memory system, a hardware register is located between the part of the computer which controls memory and the memory chips themselves. This is a device which can hold a certain amount of information at once. The register will fill up completely and then pass on all this information at once.

**24. Why is Wait states used?**

Ans: Wait states can be used to reduce the energy consumption of a processor, by allowing the main processor clock to either slow down or temporarily pause during the wait state if the CPU has no other work to do. Rather than spinning uselessly in a tight loop waiting for data, sporadically reducing the clock speed in this manner helps to keep the processor core cool and to extend battery life in portable computing devices.

**25. What is the difference between RAM and cache memory?**

Ans :RAM is the MAIN MEMORY whereas cache is a temporary memory ,frequently accessed data are stored in cache

**26. Why is RAM also known as volatile memory?**

Ans :RAM-Random Access memory  is a volatile memory because when the power cuts it have a chance to loss the memory in it.

**27. State the differences between DDR1, 2, 3 and 4**

Ans:The original DDR RAM was, very simply, exactly like the old SD RAM but with the speed at which it could transfer data doubled by transferring on both clock edges.

DDR2 RAM added a 2x clock multiplier to the module, which meant the bus clock running at the same speed as DDR RAM would be doubled, thus multiplying transfer speeds by 2 for the same bus speed.

DDR3 RAM replaces the 2x clock multiplier with a 4x clock multiplier, thus running at 4 times the memory transfer rate for the same bus speed as the original DDR RAM.

DDR4 runs on a lower voltage than DDR3, is capable of running at a higher clock speed (typical DDR3 is 1600MHz for desktop, 1333 for laptop, while typical DDR4 may be 1866 for laptop and 2133 for desktop), and can more easily come in much more dense packages (DDR3 maxes out at 16GB/DIMM slot, DDR4 *may* end up maxing out at 128GB/DIMM slot).

## ❖ CSC 307 – Operating System

**1) Explain the main purpose of an operating system?**

Operating systems exist for two main purposes. One is that it is designed to make sure a computer system performs well by managing its computational activities. Another is that it provides an environment for the development and execution of programs.

**2) What is demand paging?**

Demand paging is referred when not all of a process’s pages are in the RAM, then the OS brings the missing(and required) pages from the disk into the RAM.

**3) What are the advantages of a multiprocessor system?**

With an increased number of processors, there is a considerable increase in throughput. It can also save more money because they can share resources. Finally, overall reliability is increased as well.

**4) What is kernel?**

A kernel is the core of every operating system. It connects applications to the actual processing of data. It also manages all communications between software and hardware components to ensure usability and reliability.

**5) What are real-time systems?**

Real-time systems are used when rigid time requirements have been placed on the operation of a processor. It has well defined and fixed time constraints.

**6) What is a virtual memory?**

Virtual memory is a memory management technique for letting processes execute outside of memory. This is very useful especially is an executing program cannot fit in the physical memory.

**7) Describe the objective of multiprogramming.**

The main objective of multiprogramming is to have a process running at all times. With this design, CPU utilization is said to be maximized.

**8 ) What is time- sharing system?**

In a Time-sharing system, the CPU executes multiple jobs by switching among them, also known as multitasking. This process happens so fast that users can interact with each program while it is running.

**9) What is SMP?**

SMP is a short form of Symmetric Multi-Processing. It is the most common type of multiple-processor systems. In this system, each processor runs an identical copy of the operating system, and these copies communicate with one another as needed.

**10) What is a thread?**

A thread is a basic unit of CPU utilization. In general, a thread is composed of a thread ID, program counter, register set, and the stack.

**11) Give some benefits of multithreaded programming.**

– there is increased responsiveness to the user
– resource sharing within the process
– economy
– utilization of multiprocessing architecture

**12) Briefly explain FCFS.**

FCFS stands for First-come, first-served. It is one type of scheduling algorithm. In this scheme, the process that requests the CPU first is allocated the CPU first. Implementation is managed by a FIFO queue.

**13) What is RR scheduling algorithm?**

RR (round-robin) scheduling algorithm is primarily aimed for time-sharing systems. A circular queue is a setup in such a way that the CPU scheduler goes around that queue, allocating CPU to each process for a time interval of up to around 10 to 100 milliseconds.

**14) What are necessary conditions which can lead to a deadlock situation in a system?**

Deadlock situations occur when four conditions occur simultaneously in a system: Mutual exclusion; Hold and Wait; No preemption; and Circular wait.

**15) Enumerate the different RAID levels.**

RAID 0 – Non-redundant striping
RAID 1 – Mirrored Disks
RAID 2 – Memory-style error-correcting codes
RAID 3 – Bit-interleaved Parity
RAID 4 – Block-interleaved Parity
RAID 5 – Block-interleaved distributed Parity
RAID 6 – P+Q Redundancy

**16) What factors determine whether a detection-algorithm must be utilized in a deadlock avoidance system?**

One is that it depends on how often a deadlock is likely to occur under the implementation of this algorithm. The other has to do with how many processes will be affected by deadlock when this algorithm is applied.

**17) State the main difference between logical from physical address space.**

Logical address refers to the address that is generated by the CPU. On the other hand, physical address refers to the address that is seen by the memory unit.

**18) Give an example of a Process State.**

– New State – means a process is being created
– Running – means instructions are being executed
– Waiting – means a process is waiting for certain conditions or events to occur
– Ready – means a process is waiting for an instruction from the main processor
– Terminate – means a process is stopped abruptly

**19) What is Direct Access Method?**

Direct Access method is based on a disk model of a file, such that it is viewed as a numbered sequence of blocks or records. It allows arbitrary blocks to be read or written. Direct access is advantageous when accessing large amounts of information.

**20) What are the different types of CPU registers in a typical operating system design?**

– Accumulators
– Index Registers
– Stack Pointer
– General Purpose Registers

**21) What is the purpose of an I/O status information?**

I/O status information provides information about which I/O devices are to be allocated for a particular process. It also shows which files are opened, and other I/O device state.

**22) What is multitasking?**

Multitasking is the process within an operating system that allows the user to run several applications at the same time. However, only one application is active at a time for user interaction, although some applications can run “behind the scene”.

**23) What is caching?**

Caching is the processing of utilizing a region of fast memory for a limited data and process. A cache memory is usually much efficient because of its high access speed.

**24) What is spooling?**

Spooling is normally associated with printing. When different applications want to send an output to the printer at the same time, spooling takes all of these print jobs into a disk file and queues them accordingly to the printer.

**25) What is an Assembler?**

An assembler acts as a translator for low-level language. Assembly codes written using mnemonic commands are translated by the Assembler into machine language.

**26) What are interrupts?**

Interrupts are part of a hardware mechanism that sends a notification to the CPU when it wants to gain access to a particular resource. An interrupt handler receives this interrupt signal and “tells” the processor to take action based on the interrupt request.

**27) What is GUI?**

GUI is short for Graphical User Interface. It provides users with an interface wherein actions can be performed by interacting with icons and graphical symbols. People find it easier to interact with the computer when in a GUI especially when using the mouse. Instead of having to remember and type commands, users click on buttons to perform a process.

**28) What is preemptive multitasking?**

Preemptive multitasking allows an operating system to switch between software programs. This, in turn, allows multiple programs to run without necessarily taking complete control over the processor and resulting in system crashes.

**29) Differentiate internal commands from external commands.**

Internal commands are built-in commands that are already part of the operating system. External commands are separate file programs that are stored in a separate folder or directory.

## ❖ CSC 455 – Computer Graphics

**1. What Is Scan Conversion?**

A major task of the display processor is digitizing a picture definition given in an application program into a set of pixel-intensity values for storage in the frame buffer. This digitization process is called scan conversion.

**2. Write The Properties Of Video Display Devices?**

Properties of video display devices are persistence, resolution, and aspect ratio.

**3. What Is Rasterization?**

The process of determining the appropriate pixels for representing picture or graphics object is known as rasterization.

**4. Define Computer Graphics.**

Computer graphics remains one of the most existing and rapidly growing computer fields. Computer graphics maybe defined as a pictorial representation or graphical representation of objects in a computer.

**5. Name Any Four Input Devices?**

Four input devices are keyboard, mouse, image scanners, and trackball.

**6. Write The Two Techniques For Producing Color Displays With A Crt?**

Beam penetration method, shadow mask method.

**7. What Is Bitmap?**

Some system has only one bit per pixel; the frame buffer is often referred to as bitmap

**8. What Is Resolution?**

The maximum number of points that can be displayed without overlap on a CRT is referred to as the resolution.

**9. What Is Horizontal Retrace Of The Electron Beam?**

In raster scan display, the electron beam return to the left of the screen after refreshing each scan line, is called horizontal retrace of the electron beam.

**10. What Is Persistence?**

The time it takes the emitted light from the screen to decay one tenth of its original intensity is called as persistence.

**11. What Is Aspect Ratio?**

The ratio of vertical points to the horizontal points necessary to produce length of lines in both directions of the screen is called the Aspect ratio. Usually the aspect ratio is ¾.

**12. Define Pixel?**

Pixel is shortened forms of picture element. Each screen point is referred to as pixel or pel.

**13. What Is Frame Buffer?**

Picture definition is stored in a memory area called frame buffer or refresh buffer.

**14. What Is Point In The Computer Graphics System?**

The point is a most basic graphical element & is completely defined by a pair of user coordinates (x, y).

**15. Define Circle?**

Circle is defined by its center xc, yc and its radius in user coordinate units. The equation of the circle is (x-xc) + (yyc)= r2.

**15. What Is Transformation?**

Transformation is the process of introducing changes in the shape size and orientation of the object using scaling rotation reflection shearing & translation etc.

**16. What Is Translation?**

Translation is the process of changing the position of an object in a straight-line path from one coordinate location to another. Every point (x , y) in the object must under go a displacement to (x|,y|). the transformation is:x| = x + tx ; y| = y+ty

**17. What Is Rotation?**

A 2-D rotation is done by repositioning the coordinates along a circular path, in the x-y plane by making an angle with the axes. The transformation is given by:X| = r cos (q + f) and Y| = r sin (q + f).

**18. What Is Scaling?**

A 2-D rotation is done by repositioning the coordinates along a circular path, in the x-y plane by making an angle with the axes. The transformation is given by:X| = r cos (q + f) and Y| = r sin (q + f).

**19. What Is Shearing?**

The shearing transformation actually slants the object along the X direction or the Y direction as required. ie; this transformation slants the shape of an object along a required plane.

**20. What Is Reflection?**

The reflection is actually the transformation that produces a mirror image of an object. For this use some angles and lines of reflection.

**21. What Are The Two Classifications Of Shear Transformation?**

X shear, y shear.
A point (4,3) is rotated counterclockwise by an angle of45°. Find the rotation matrix and the resultant point

**22. What Is Computer Graphics?**

The term computer graphics include almost everything on computer that is not text or sound. It is an art of drawing pictures, lines, charts, etc. using computers with the help of programming. Or we can say that graphics is the representation and manipulation of image data by computer with the help from specialized software and hardware. Graphic designing is done using the various available soft wares for computers which can produce the 3D images in the required shape and dimension. Computer graphics help us in getting the real display experiences.

**23. What Are The Raster And Vector Graphics?**

The Raster and Vector graphics can be explained as-

**RASTER**- In computer graphics image, or BITMAP, is a dot matrix data structure representing a generally rectangular grid of pixels or points of color, viewable via a monitor, paper, or other display medium. Raster image are stored in image files with varying formats.

**VECTOR**- Vector graphics is the use of geometrical primitives such as points, lines, curves, and shapes or polygon, which are all based on mathematical expressions, to represent image in computer graphics. “Vector”, in this context, implies more than a straight line.

### 24. Define Random and Raster Scan Displays?

Random scan is a method in which display is made by electronic beam, which is directed only to the points or parts of the screen where picture is to be drawn.

The Raster scan system is a scanning technique in which the electron sweeps from top to bottom and from left to right. The intensity is turned on or off to light and un-light the pixel.

## ❖ CSC 437 – Compiler Design

### 1. What Is A Compiler?

A compiler is a program that reads a program written in one language –the source language and translates it into an equivalent program in another language-the target language. The compiler reports to its user the presence of errors in the source program.

### 2. What Are The Two Parts Of A Compilation? Explain Briefly.

Analysis and Synthesis are the two parts of compilation.

The analysis part breaks up the source program into constituent pieces and creates an intermediate representation of the source program.
The synthesis part constructs the desired target program from the intermediate representation.

### 3. What Are The Various Ways To Pass A Parameter In A Function?

- Call by value
- Call by reference
- Copy-restore
- Call by name

### 4. List The Different Storage Allocation Strategies.

The strategies are:

- Static allocation
- Stack allocation
- Heap allocation

### 5. What Are The Contents Of Activation Record?

The activation record is a block of memory used for managing the information needed by a single execution of a procedure. Various fields f activation record are:

- Temporary variables
- Local variables
- Saved machine registers
- Control link
- Access link
- Actual parameters
- Return values

**6. Define Symbol Table.**

Symbol table is a data structure used by the compiler to keep track of semantics of the variables. It stores information about scope and binding information about names.

**7. What Is Linear Analysis?**

Linear analysis is one in which the stream of characters making up the source program is read from left to right and grouped into tokens that are sequences of characters having a collective meaning.Also called lexical analysis or scanning.

**8. List The Various Phases Of A Compiler.**

The following are the various phases of a compiler:

- Lexical Analyzer
- Syntax Analyzer
- Semantic Analyzer
- Intermediate code generator
- Code optimizer
- Code generator

**9. Differentiate Tokens, Patterns, and Lexeme.**

**Tokens-** Sequence of characters that have a collective meaning.
**Patterns-** There is a set of strings in the input for which the same token is produced as output. This set of strings is described by a rule called a pattern associated with the token
**Lexeme-** A sequence of characters in the source program that is matched by the pattern for a token.

**10. Write A Regular Expression For An Identifier.**

An identifier is defined as a letter followed by zero or more letters or digits.
The regular expression for an identifier is given as
**letter (letter | digit)***

**11. Define A Context Free Grammar.**

A context free grammar G is a collection of the following
· V is a set of non-terminals
· T is a set of terminals
· S is a start symbol
· P is a set of production rules
G can be represented as G = (V,T,S,P)
Production rules are given in the following form
Non terminal → (V U T)*

**12. Briefly Explain The Concept Of Derivation.**

Derivation from S means generation of string w from S. For constructing derivation two things are important.
i) Choice of non-terminal from several others.
ii) Choice of rule from production rules for corresponding non terminal.
Instead of choosing the arbitrary non terminal one can choose
i) either leftmost derivation – leftmost non terminal in a sentinel form.
ii) or rightmost derivation – rightmost non terminal in a sentinel form.

**13. Define Ambiguous Grammar.**

A grammar G is said to be ambiguous if it generates more than one parse tree for some sentence of language L(G).
i.e. both leftmost and rightmost derivations are same for the given sentence.

**14. What Are The Various Types Of Intermediate Code Representation?**

There are mainly three types of intermediate code representations.

1. Syntax tree
2. Postfix
3. Three address code

## ❖ CSC 469 – Software Engineering

**1. What is computer software?**

Computer software is a complete package, which includes software program, its documentation and user guide on how to use the software.

**2. Can you differentiate computer software and computer program?**

A computer program is piece of programming code which performs a well-defined task whereas software includes programming code, its documentation and user guide.

**3. What is software engineering?**

Software engineering is an engineering branch associated with software system development.

**4. When you know programming, what is the need to learn software engineering concepts?**

A person who knows how to build a wall may not be good at building an entire house. Likewise, a person who can write programs may not have knowledge of other concepts of Software Engineering. The software engineering concepts guide programmers on how to assess requirements of end user, design the algorithms before actual coding starts, create programs by coding, testing the code and its documentation.

**5. What is software process or Software Development Life Cycle (SDLC)?**

Software Development Life Cycle, or software process is the systematic development of software by following every stage in the development process namely, Requirement Gathering, System Analysis, Design, Coding, Testing, Maintenance and Documentation in that order.

**6. What are SDLC models available?**

There are several SDLC models available such as Waterfall Model, Iterative Model, Spiral model, V-model and Big-bang Model etc.

**7. What are various phases of SDLC?**

The generic phases of SDLC are: Requirement Gathering, System Analysis and Design, Coding, Testing and implementation. The phases depend upon the model we choose to develop software.

**8. What is software project management?**

Software project management is process of managing all activities like time, cost and quality management involved in software development.

**9. Who is software project manager?**

A software project manager is a person who undertakes the responsibility of carrying out the software project.

**8. What does software project manager do?**

Software project manager is engaged with software management activities. He is responsible for project planning, monitoring the progress, communication among stakeholders, managing risks and resources, smooth execution of development and delivering the project within time, cost and quality contraints.

**10. What is software scope?**

Software scope is a well-defined boundary, which encompasses all the activities that are done to develop and deliver the software product.

The software scope clearly defines all functionalities and artifacts to be delivered as a part of the software. The scope identifies what the product will do and what it will not do, what the end product will contain and what it will not contain.

**11. What is project estimation?**

It is a process to estimate various aspects of software product in order to calculate the cost of development in terms of efforts, time and resources. This estimation can be derived from past experience, by consulting experts or by using pre-defined formulas.

**12. What are function points?**

Function points are the various features provided by the software product. It is considered as a unit of measurement for software size.

**13. What is change control?**

Change control is function of configuration management, which ensures that all changes made to software system are consistent and made as per organizational rules and regulations.

**14. Mention some project management tools.**

There are various project management tools used as per the requirements of software project and organization policies. They include Gantt Chart, PERT Chart, Resource Histogram, Critical Path Analysis, Status Reports, Milestone Checklists etc.

**15. What are software requirements?**

Software requirements are functional description of proposed software system. Requirements are assumed to be the description of target system, its functionalities and features. Requirements convey the expectations of users from the system.

**16. How can you gather requirements?**

Requirements can be gathered from users via interviews, surveys, task analysis, brainstorming, domain analysis, prototyping, studying existing usable version of software, and by observation.

**17. What is SRS?**

SRS or Software Requirement Specification is a document produced at the time of requirement gathering process. It can be also seen as a process of refining requirements and documenting them.

**18. What are functional requirements?**

Functional requirements are functional features and specifications expected by users from the proposed software product.

**19. What are non-functional requirements?**

Non-functional requirements are implicit and are related to security, performance, look and feel of user interface, interoperability, cost etc.

**20. Mentions some software analysis & design tools?**

These can be: DFDs (Data Flow Diagrams), Structured Charts, Structured English, Data Dictionary, HIPO (Hierarchical Input Process Output) diagrams, ER (Entity Relationship) Diagrams and Decision tables.

**21. What is level-0 DFD?**

Highest abstraction level DFD is known as Level 0 DFD also called a context level DFD, which depicts the entire information system as one diagram concealing all the underlying details.

**22. What is the difference between structured English and Pseudo Code?**

Structured English is native English language used to write the structure of a program module by using programming language keywords, whereas, Pseudo Code is more close to programming language and uses native English language words or sentences to write parts of code.

**23. What is data dictionary?**

Data dictionary is referred to as meta-data. Meaning, it is a repository of data about data. Data dictionary is used to organize the names and their references used in system such as objects and files along with their naming conventions.

**24. What is the difference between function oriented and object oriented design?**

Function-oriented design is comprised of many smaller sub-systems known as functions. Each function is capable of performing significant task in the system. Object oriented design works around the real world objects (entities), their classes (categories) and methods operating on objects (functions).

**25. Briefly define top-down and bottom-up design model.**

Top-down model starts with generalized view of system and decomposes it to more specific ones, whereas bottom-up model starts with most specific and basic components first and keeps composing the components to get higher level of abstraction.

**26. Differentiate validation and verification?**

Validation checks if the product is made as per user requirements whereas verification checks if proper steps are followed to develop the product.

Validation confirms the right product and verification confirms if the product is built in a right way.

**27. What is black-box and white-box testing?**

Black-box testing checks if the desired outputs are produced when valid input values are given. It does not verify the actual implementation of the program.

White-box testing not only checks for desired and valid output when valid input is provided but also it checks if the code is implemented correctly.

| Criteria | Black Box Testing | White Box Testing |
|---|---|---|
| Knowledge of software program, design and structure essential | No | Yes |

| | | |
|---|---|---|
| Knowledge of Software Implementation essential | No | Yes |
| Who conducts this test on software | Software Testing Employee | Software Developer |
| baseline reference for tester | Requirements specifications | Design and structure details |

**28. Quality assurance vs. Quality Control?**

Quality Assurance monitors to check if proper process is followed while software developing the software.

Quality Control deals with maintaining the quality of software product.

**29. What are various types of software maintenance?**

Maintenance types are: corrective, adaptive, perfective and preventive.

- **Corrective**

  Removing errors spotted by users

- **Adaptive**

  tackling the changes in the hardware and software environment where the software works

- **Perfective maintenance**

  implementing changes in existing or new requirements of user

- **Preventive maintenance**

  taking appropriate measures to avoid future problems

### 30. What is software re-engineering?

Software re-engineering is process to upgrade the technology on which the software is built without changing the functionality of the software. This is done in order to keep the software tuned with the latest technology.

### 31. What are CASE tools?

CASE stands for Computer Aided Software Engineering. CASE tools are set of automated software application programs, which are used to support, accelerate and smoothen the SDLC activities.