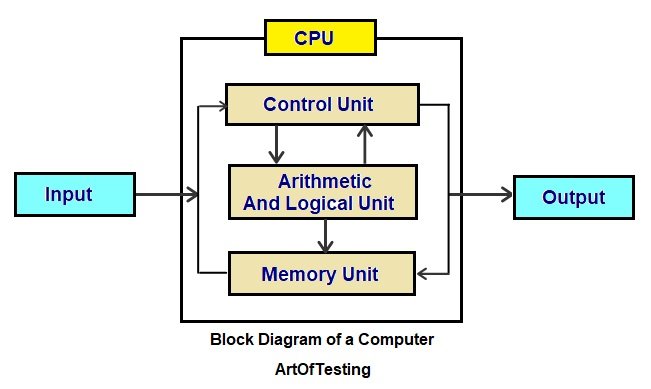
**IMPORTANT QUESTIONS**

**THEORY:**

1. What is Computer? Explain the block diagram of the computer.

Ans:



.

**The Central Processing Unit (CPU)**

The CPU can be called the brain of a computer system.

CPU perform all types of data processing operations.It stores data,intermediate results and instructions(program).It controls operations of all parts of computer.

CPU itself has the following 3 components.

**(a)Memory Unit**

**(b)Control Unit (CU)**

**(c)Arithmetic Logic Unit (ALU)**

**INPUT UNIT**

**Input unit consists of devices with the help of which we enter the information to the computer.This unit is a link between user and computer.**

**Memory Unit**

[Memory](https://codescracker.com/computer-fundamental/computer-memory-and-types.htm) units are an integral part of any modern digital computer. It is the repository for all of the results, both intermediate and final.

The data that are read from the primary storage or an input unit are moved to the memory of the computer so that they can be processed. These data can come from either the main storage or an input unit.

The data that needs to be processed and the instructions that need to be carried out are both stored in this memory unit so they can be accessed quickly.

**Functions of Memory Unit are**

a)It stores all the data to be processes and the instructions required for processing.

b)It stores intermediate results of processing.

c) It stores final results of processing before these result are released to an output device.

d)All input and outputs are transmitted through main memory.

**Disk Storage Unit**

Primary and secondary storage units are the two categories that can be found in a storage facility. Now let's briefly define these two storage units one by one, starting with the "primary storage unit."

**Primary Storage Unit**

[Primary memory](https://codescracker.com/computer-fundamental/computer-memory-and-types.htm#b) is connected to the input unit as well as the output unit in a straightforward manner. It stores both the data that was input and the result of the calculation.

**Secondary Storage Unit**

It is not possible to store data permanently on the primary storage for use at a later time. Because of this, additional forms of data storage technology, also known as [secondary or auxiliary storage](https://codescracker.com/computer-fundamental/computer-memory-and-types.htm#a), are required in order to store the data in a manner that is both permanent and accessible over an extended period of time.

**Control Unit (CU)**

In addition, the control unit of a central processing unit is responsible for directing the overall operation of a computer. In addition to this, it exercises control over all devices connected to the CPU, including memory and input/output devices.

The CU is responsible for retrieving instructions from memory, decoding those instructions, interpreting those instructions to determine what tasks are to be carried out, and then sending appropriate control signals to the other components so that they can carry out the steps necessary to execute the instruction.

**Arithmetic Logic Unit (ALU)**

The arithmetic logic unit is responsible for carrying out all of the mathematical operations, including addition, subtraction, multiplication, and division. In addition to that, a logical operation is used for the comparison.

**OUTPUT UNIT**

Output unit **consists of devices with the help of which we get the information from computer.This unit is a link between computer and user.**

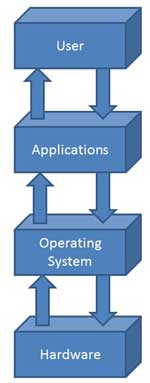
1. **What is Operating System? Write its characteristics, objectives and types.**

Operating System

* An operating system is a program that acts as an interface between the software and the computer hardware.
* It is an integration set of specialised programs that are used to manage overall resources and operations of the computer.
* It is a specialised software that controls and monitors the execution of all other programs that reside in the computer, including application programs and other system software.

Objectives of Operating System

* Making a computer system convenient to use in an efficient manner.
* To hide the details of the hardware resources from the users.
* To provide users a convenient interface to use the computer system.
* To act as an intermediary between the hardware and its users and making it easier for the users to access and use other resources.
* Manage the resources of a computer system.
* Keep track of who is using which resource, granting resource requests, according for resource using and mediating conflicting requests from different programs and users.
* The efficient and fair sharing of resources among users and programs.



Characteristics of Operating System

* **Memory Management** -- It keeps track of primary memory, i.e., what parts of it are in use by whom, what parts are not in use, etc. Allocates the memory when the process or program requests it.
* **Processor Management** -- Allocates the processor (CPU) to a process. Deallocate processor when processor is no longer required.
* **Device Management** -- Keeps tracks of all devices. This is also called I/O controller. Decides which process gets the device when and for how much time.
* **File Management** -- Allocates the resources. Deallocates the resource. Decides who gets the resources.
* **Security** -- By means of passwords & similar other techniques, preventing unauthorized access to programs & data.
* **Job accounting** -- Keeping track of time & resources used by various jobs and/or users.
* **Control over system performance** -- Recording delays between requests for a service & from the system.
* **Interaction with the operators** -- The interaction may take place via the console of the computer in the form of instructions. Operating System acknowledges the same, do the corresponding action and inform the operation by a display screen.
* **Error-detecting aids** -- Production of dumps, traces, error messages and other debugging and error-detecting methods.
* **Coordination between other software and users** -- Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems

1. **What is Software and Hardware? Explain different types of software.**

Software is a set of programs, which are designed to perform a well-defined function. A program is a sequence of instructions written to solve a particular problem.

There are two types of softwares:

* System Software
* Application Software

## System Software

* The system software is a collection of programs designed to operate, control and extend the processing capabilities of the computer itself. System software are generally prepared by computer manufacturers.
* These softwares comprise of programs written in low level languages which interact with the hardware at a very basic level. System software serves as the interface between hardware and the end users.
* Some examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc.
* Features of System Software are the following:
* Close to system.
* Fast in speed.
* Difficult to design.
* Difficult to understand.
* Less interactive.
* Smaller in size.
* Difficult to manipulate.
* Generally written in low-level language.

## Application Software

* Application softwares are the softwares that are designed to satisfy a particular need of a particular environment. All softwares prepared by us in the computer lab can come under the category of Application Software.
* Application software may consist of a single program, such as a Microsoft's notepad for writing and editing simple text. It may also consist of a collection of programs, often called a software package, which work together to accomplish a task, such as a spreadsheet package.
* Examples of Application softwares are the following:
* Payroll Software
* Student Record Software
* Inventory Management Software
* Income Tax Software
* Railways Reservation Software
* Microsoft Office Suite Software
* Microsoft Word
* Microsoft Excel
* Microsoft Powerpoint

1. **Explain different types of computer.**

**Computer – Types**

Computers can be broadly classified by their speed and computing power.

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Type** | **Specifications** |
| 1 | PC (Personal Computer) | Single user computer system. Moderately powerful microprocessor. |
| 2 | WorkStation | Single user computer system. Similar to Personal Computer but have more powerful microprocessor. |
| 3 | Mini Computer | Multi-user computer system. Capable of supporting hundreds of users simultaneously. |
| 4 | Main Frame | Multi-user computer system. Capable of supporting hundreds of users simultaneously. Software technology is different from minicomputer. |
| 5 | Supercomputer | An extremely fast computer, which can perform hundreds of millions of instructions per second. |

PC (Personal Computer)

A PC can be defined as a small, relatively inexpensive computer designed for an individual user. PCs are based on the microprocessor technology that enables manufacturers to put an entire CPU on one chip. Businesses use personal computers for word processing, accounting, desktop publishing, and for running spreadsheet and database management applications. At home, the most popular use for personal computers is for playing games and surfing the Internet.

Although personal computers are designed as single-user systems, these systems are normally linked together to form a network. In terms of power, nowadays high-end models of the Macintosh and PC offer the same computing power and graphics capability as low-end workstations by Sun Microsystems, Hewlett-Packard, and DELL.

## WorkStation

Workstation is a computer used for engineering applications (CAD/CAM), desktop publishing, software development, and other such types of applications, which require a moderate amount of computing power and relatively high quality graphics capabilities.

Workstations generally come with a large, high-resolution graphics screen, large amount of RAM, inbuilt network support, and a graphical user interface. Most workstations also have a mass storage device such as a disk drive, but a special type of workstation, called a diskless workstation, comes without a disk drive.

Common operating systems for workstations are UNIX and Windows NT. Like PC, Workstations are also single-user computers. However, workstations are typically linked together to form a local-area network, although they can also be used as stand-alone systems.

## Minicomputer

It is a midsize computer. A minicomputer is a multi-processing system capable of supporting from up to 250 users simultaneously

## Mainframe

Mainframe is a very large in size and an expensive computer capable of supporting hundreds, or even thousands, of users simultaneously. Mainframe executes many programs concurrently. Mainframes support many simultaneous programs execution

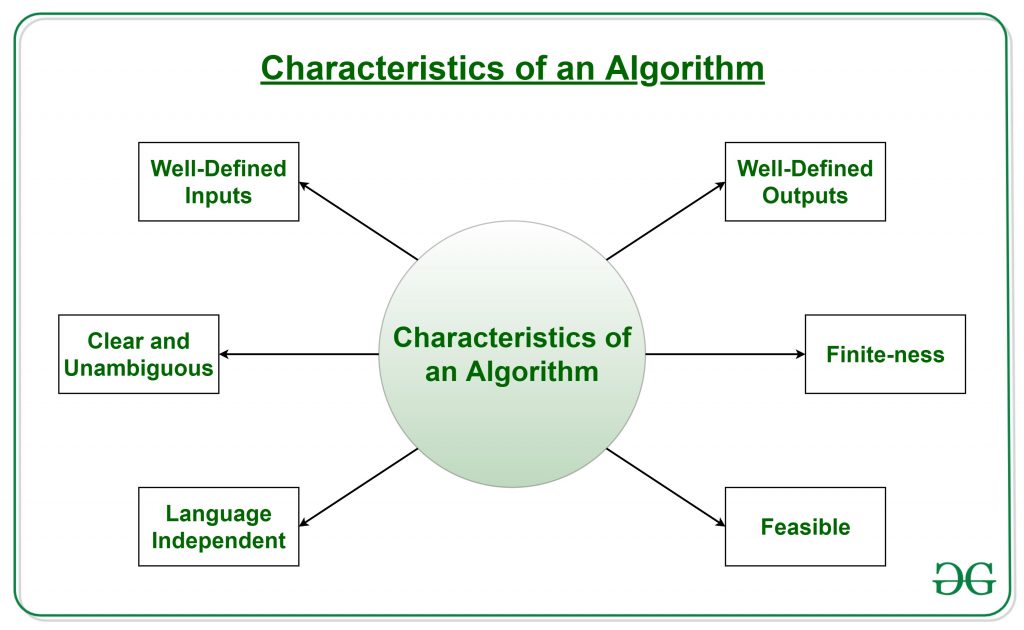
## Supercomputer

Supercomputers are one of the fastest computers currently available. Supercomputers are very expensive and are employed for specialized applications that require immense amounts of mathematical calculations (number crunching). For example, weather forecasting, scientific simulations, (animated) graphics, fluid dynamic calculations, nuclear energy research, electronic design, and analysis of geological data (e.g. in petrochemical prospecting).

1. What is algorithm? Write the characteristics of algorithm.

Ans:

*The word*[***Algorithm***](https://www.geeksforgeeks.org/fundamentals-of-algorithms/)*means ”*Algorithm is a step by step procedure to solve any particular problems *”*



* **Input**: An algorithm has zero or more inputs.
* **Output**: An algorithm produces at least one output.
* **Definiteness:** All instructions in an algorithm must be unambiguous, precise, and easy to interpret. By referring to any of the instructions in an algorithm one can clearly understand what is to be done. Every fundamental operator in instruction must be defined without any ambiguity.
* **Finiteness:** An algorithm must terminate after a finite number of steps in all test cases. Every instruction which contains a fundamental operator must be terminated within a finite amount of time.
* **Effectiveness:**An algorithm must be developed by using very basic, simple, and feasible operations so that one can trace it out by using just paper and pencil.

6 Memory hierarchy of the computer?

Ans The Computer memory hierarchy looks like a pyramid structure which is used to describe the differences among [**memory types**](https://www.tutorialspoint.com/computer_fundamentals/computer_memory.htm). It separates the computer storage based on hierarchy.

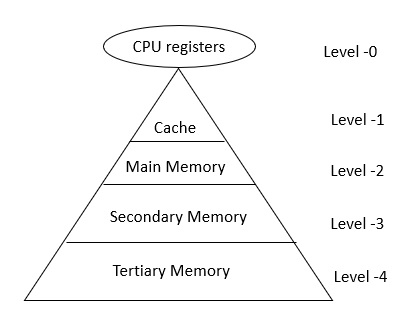
Level 0: CPU registers

Level 1: Cache memory

Level 2: Main memory or primary memory

Level 3: Magnetic disks or secondary memory

Level 4: Optical disks or magnetic types or tertiary Memory



In Memory Hierarchy the cost of memory, capacity is inversely proportional to speed. Here the devices are arranged in a manner Fast to slow, that is form register to Tertiary memory.

Let us discuss each level in detail:

Level-0 − Registers

The registers are present inside the [**CPU**](https://www.tutorialspoint.com/computer_fundamentals/computer_cpu.htm). As they are present inside the CPU, they have least access time. Registers are most expensive and smallest in size generally in kilobytes. They are implemented by using Flip-Flops.

Level-1 − Cache

[**Cache memory**](https://www.tutorialspoint.com/what-is-cache-memory-functions-and-types-of-cache-memory) is used to store the segments of a program that are frequently accessed by the processor. It is expensive and smaller in size generally in Megabytes and is implemented by using static [**RAM**](https://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm).

Level-2 − Primary or Main Memory

It directly communicates with the CPU and with auxiliary memory devices through an I/O processor. [**Main memory**](https://www.tutorialspoint.com/what-is-the-main-memory) is less expensive than cache memory and larger in size generally in Gigabytes. This memory is implemented by using [**dynamic RAM**](https://www.tutorialspoint.com/Dynamic-Random-Access-Memory-DRAM).

Level-3 − Secondary storage

[**Secondary storage devices**](https://www.tutorialspoint.com/basics_of_computers/basics_of_computers_secondary_memory.htm) like [**Magnetic Disk**](https://www.tutorialspoint.com/what-are-magnetic-disks) are present at level 3. They are used as backup storage. They are cheaper than main memory and larger in size generally in a few TB.

Level-4 − Tertiary storage

Tertiary storage devices like magnetic tape are present at level 4. They are used to store removable files and are the cheapest and largest in size (1-20 TB).

1. Define all:
2. **Keyword**
3. **Comment**
4. **Datatype**

(a)A keyword is a **reserved word**. You cannot use it as a variable name, constant name, etc. There are only 32 reserved words (keywords) in the C language.

A list of 32 keywords in the c language is given below:

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

We will learn about all the C language keywords later.

**(b)Comment**

Comment is that which inactivate any line in the program. There two type of comment.

1. Single line comment(//)

It is that which inactive any one line.

**Example**

// x=y;

//y=z;

//Z=v;

1. Multiline comment(/\*………………….\*/)

It is that which inactive one or more than one line.

/\*

X=y;

Y=z;

Z=v;

\*/

**©Data type**

Data type is that which reserve the size inside the memory of any variable. There is no other method to reserve the size inside the memory.

**Data type**

**Primitive Non primitive or user define or derive type**

Numeric non numeric Array,struct etc

Int,double char

| **Type** | **Size (bytes)** | **Format Specifier** |
| --- | --- | --- |
| int | at least 2, usually 4 | %d, %i |
| char | 1 | %c |
| float | 4 | %f |
| double | 8 | %lf |
| short int | 2 usually | %hd |
| unsigned int | at least 2, usually 4 | %u |
| long int | at least 4, usually 8 | %ld, %li |
| long long int | at least 8 | %lld, %lli |
| unsigned long int | at least 4 | %lu |
| unsigned long long int | at least 8 | %llu |
| signed char | 1 | %c |
| unsigned char | 1 | %c |
| long double | at least 10, usually 12 or 16 | %Lf |
| **Type** | **Range** | **Size (in bytes)** |
| unsigned char | 0 to 255 | 1 |
| signed char or char | -128 to +127 | 1 |
| unsigned int | 0 to 65535 | 2 |
| signed int or int | -32,768 to +32767 | 2 |
| unsigned short int | 0 to 65535 | 2 |
| signed short int or short int | -32,768 to +32767 | 2 |
| unsigned long int | 0 to +4,294,967,295 | 4 |
| signed long int or long int | -2,147,483,648 to +2,147,483,647 | 4 |
| long double | 3.4E-4932 to 1.1E+4932 | 10 |
| double | 1.7E-308 to 1.7E+308 | 8 |
| float | 3.4E-38 to 3.4E+38 | 4 |

(d)Statement

There is 4 types of statement :-

1. Single statement :- Single line statement or expression statement contain only one line. EX

x=y;

y=z;

v=x;

ii Compound statement :- Which has more than one line writhen a block { }.

EX

{

x=y;

y=z;

v=x;

}

* 1. Control statement :- Control statement is that which control any block or program.

EX

If(x>y)

{

x=y;

y=z;

v=x;

}

* 1. Jumping statement :- Jumping statement is that which jump from one place to another place.

EX break, continue, gotoxy(), goto.

1. What is storage class? Define all storage class with example.

**Storage Class-Storage class is that which store the data In a special memory which can be access fastly**

**Types of Storage Classes in C**

There are four different types of storage classes that we use in the C language:

* **Automatic Storage Class**
* **External Storage Class**
* **Static Storage Class**
* **Register Storage Class**

**Use of Storage Class in C**

A variable given in a C program will have two of the properties: storage class and type. Here, type refers to any given variable’s data type, while the storage class determines that very variable’s lifetime, visibility, and also its scope.

**Summary of Storage Classes in C**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Class** | **Name of Class** | **Place of Storage** | **Scope** | **Default Value** | **Lifetime** |
| auto | Automatic | memory | Local | Garbage Value | As long as the program control remains within the block in which it is declared. |
| extern | External | memory | Global | Zero | Till the main program ends. One can declare it anywhere in a program. |
| static | Static | memory | Local | Zero | Till the main program ends. It contain the current value. |
| register | Register | Register | Local | Garbage Value | As long as the program control remains within the block in which it is declared |

Example auto or static or register

Void main()

{

auto int x=10;

printf(“x=%d”,x);

}

Example of extern:

extern int x=10;

1. What is precedence and associatively?

Ans:**Operator Precedence and Associativity Table**

Precedence is the priority for grouping different types of operators with their operands. Associativity is the left-to-right or right-to-left order for grouping operands to operators that have the same precedence.

**For example**, in the expression 1 + 5 \* 3 , the answer is 16 and not 18 because the multiplication ("\*") operator has a higher precedence than the addition ("+") operator. Parentheses may be used to force precedence, if necessary.

The following tables list the C operator precedence from highest to lowest and the associativity for each of the operators:

| **Precedence** | **Operator** | **Description** | **Associativity** |
| --- | --- | --- | --- |
| 1 | **()** | Parentheses (function call) | Left-to-Right |
| **[]** | Array Subscript (Square Brackets) |
| **.** | Dot Operator |
| **->** | Structure Pointer Operator |
| **++ , —** | Postfix increment, decrement |
| 2 | **++ / —** | Prefix increment, decrement | Right-to-Left |
| **+ / –** | Unary plus, minus |
| **! , ~** | Logical NOT,  Bitwise complement |
| **(type)** | Cast Operator |
| **\*** | Dereference Operator |
| **&** | Addressof Operator |
| **sizeof** | Determine size in bytes |
| 3 | **\*,/,%** | Multiplication, division, modulus | Left-to-Right |
| 4 | **+/-** | Addition, subtraction | Left-to-Right |
| 5 | **<< , >>** | Bitwise shift left, Bitwise shift right | Left-to-Right |
| 6 | **< , <=** | Relational less than, less than or equal to | Left-to-Right |
| **> , >=** | Relational greater than, greater than or equal to |
| 7 | **== , !=** | Relational is equal to, is not equal to | Left-to-Right |
| 8 | **&** | Bitwise AND | Left-to-Right |
| 9 | **^** | Bitwise exclusive OR | Left-to-Right |
| 10 | **|** | Bitwise inclusive OR | Left-to-Right |
| 11 | **&&** | Logical AND | Left-to-Right |
| 12 | **||** | Logical OR | Left-to-Right |
| 13 | **?:** | Ternary conditional | Right-to-Left |
| 14 | **=** | Assignment | Right-to-Left |
| **+= , -=** | Addition, subtraction assignment |
| **\*= , /=** | Multiplication, division assignment |
| **%= , &=** | Modulus, bitwise AND assignment |
| **^= , |=** | Bitwise exclusive, inclusive OR assignment |
| **<<=, >>=** | Bitwise shift left, right assignment |
| 15 | **,** | comma (expression separator) | Left-to-Right |

1. Write the difference between: a. RAM & ROM

b. Compiler & interpreter

c. while & do-while

d. if-else & switch

Ans (a). RAM :-

1. Volatile
2. Read and write operation supported.
3. Expensive
4. Temporary storage.
5. Requires flow of electricity to retain data

ROM :-

1. Non-volatile

ii Read only operation supported.

Iii Cheap not expensive

Iv Permanent stores

V Does not require flow of electricity to retain data.

b). Complier :-

1. It can compile whole program at the time.
2. More memory required
3. Compiler can compile conditional statement faster than interpreter.
4. Compiler show the error in one time.
5. Turbo C, Turbo C2
6. Faster
7. Suitable for large programs.
8. It create and stores an object file.
9. More useful in case of security.

Interpreter :-

1. It can compile any program step by step
2. Less memory required
3. Interpreter can compile conditional statement less than compile.
4. Interpreter does not show the error in one time
5. Basic, Foxpro
6. Slower
7. Suitable for small programs
8. It does not create an object program.
9. Not more useful in case of security.

c). While:-

1. Entry control loop
2. First checking the condition if true then do the block work
3. Not terminated by (;) semicolon
4. Do not use do
5. Syntax :

**while (cond)**

{

Statement 1 ;

Statement 2 ;

Statement n;

inc or dec or update;

}

Do while :-

1. Exist control loop
2. Without checking the condition in the first time must do the block work
3. Must terminated by **(;)**
4. Use **do**
5. Syntax :

**do**

{

stat 1;

stat2;

stat n;

inc or dec or update

} **while (cond);**

d). if-else :-

1. It can check only one case not multiple case
2. We pass condition also.
3. Use else
4. Not use case
5. If(condition)

{

Statement 1;

}

else

{

Statement 2;

}

Switch() :-

* 1. We can check multiple case in one
  2. We pass expression not condition
  3. Use default
  4. Use case
  5. Switch (exp)

{

case 1 : statement 1;

break;

case 2 : statement 2;

break;

……………………

………………………………

case n: statement n;

break;

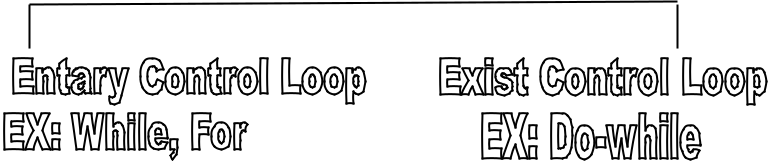
default: statement;

}

1. What is loop? Define all loops with syntax and example.(while,do-while,for)

Ans. Loop: - Loop means rotating any block till the condition is true

Loop



* While is a loop which is used for rotating any block. In this loop first checking the condition, If condition is true then go to the block and do all the statement. In the last statement do the increment or decrement or update work. After that again checking the condition. If true then again do all the above work. Otherwise comeback from the loop.

Syntax:-

While (condition)

{

Statement 1;

statement2;

……..

Statement n;

Increment or dec or update;

}

* Do-while is a loop which is used for rotating any block. In this loop first do all the statement of the block.In second time checking the condition If condition is true then again do the blocks work. Otherwise comeback from the loop.

Syntax:-

do

{

Statement 1;

statement2;

……..

Statement n;

inc or dec or update;

}

while (cond)**;**

* for :- For is a loop which used for rotating any block. In this loop has three parts
  1. Initialization

1. Condition
2. Increment and decrement

All these part separate by semicolon. In this loop first Initialize the variable after that checking the

condition. If true then go to the block and do all the statement. After that do the increment and

decrement and again checking the condition if true then again do the block work otherwise comeback from the loop.

**Syntax :-**

for(Initialization ; condtion; inc or dec)

{

Statement 1;

statement2;

……..

Statement n;

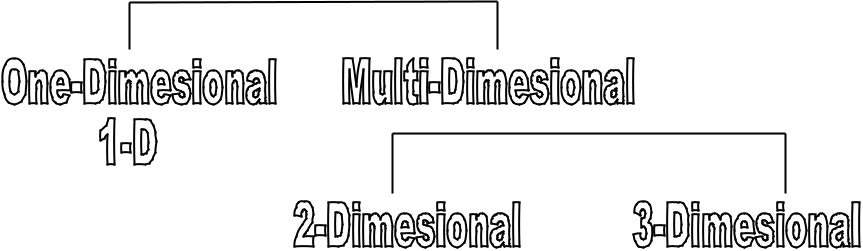
}

1. What is array? Define different type of array.

Ans. Array :- Array is that which contain more than one value in a one variable all the value are homogeneous. In array we can store the value by sing index. Index starting from is 0 and goes tp n-1.

* Array are two type :-

Array



* + **One-Dimensional** :- In one dimensional array if has one parameter or subscript.

EX

Int a [10];

In this example variable ‘a’ is integer type it has capacity to store 10 integer value.

In another example

Example

char a[10];

In this example variable ‘a’ is a character type it has capacity to store only 9 value not 10, the last index value is always Null or ‘\0’.

* + **2-Dimensional Array** :- In 2-dimensional array it has 2 parameter or subscript one is row and second is column.

EX int a[2] [3];

In this example variable has two parameter one is Row i.e 2 and second one is column i.e 3. It has capacity to store 2\*3=6 integer value, and the size of the array is 6\*2=12bytes.