## Day1 :

### 1. Source code vs Object Code:

**Source code** : The original file which is written by the programmer.
**Eg**: Hello.java, sum.c..etc

**Object code** : It is the machine executable file having instructions for the machine in the form of binary digits. And this code is generated by the compiler.
**Eg**: Hello.class, sum.obj.

### 2. Using compiler with flags

The selection of the compiler flags can optimize the performance of specific types of applications.
Eg : -g : Generate debugging option
-O2 :Recommended optimizations
-wall : Recommended compiler warnings

### 3. Keywords and Variables

**Keywords :** Keywords are reserved by the program because those words have a special meaning. Every programming language has a set of keywords that cannot be used as variable names.

**Variables :** Variables are used to store information to be referenced and manipulated in a program.

### 4. Variable Scope, Lifetime, Initial Value, Modifiers

**Variable Scope :** It refers to the extent of the code in which a variable can be accessed and modified.
**Eg :** Class level variables, Method level variables, local variables etc

**Lifetime :** The time during which the variable stays in memory and therefore accessible during program execution.

**Eg:** Local variables are created when a method is called and destroyed after the execution of that method.

**Initial Value :** It is the value a variable starts with or initially set to.
**Eg:** int a;
Here by default initial value of a is 0.

**Modifiers :** These are the specifiers that are appended to the beginning of class, variables, methods and constructors in order to control their accessibility and other functionalities.

Access Modifiers-controls the access level. Eg: public,private etc

Non-Access Modifies- Do not control the access level, but provides other functionality. Eg: final,static

## 5. Programming Constructs(Selection constructs and loop constructs)

Programs are designed using common building blocks. These building blocks are known as programming constructs(programming concepts). There are 3 building blocks to consider :
**1.Sequence :** It is the order in which the programming statements are executed.
**2.Selection :** It is a programming construct where a section of code is run only if a condition is met.
**3.Iteration(Loop) :** It is the repeated execution of a section of code when a program is running.

## 6. Declare Array

An array declaration names the array, specifies the type of elements and define the number of elements in the array. A variable with array type is considered as a pointer to the type of the array elements.

Eg: int[] arr=new int[10];

## 7. Store data in array and process elements

```
int a[]=new int[3];
a[0]=10;
a[1]=20;
for(int i=0;i<3;i++)
{
```

```
        System.out.println(a[i]+i);
}
```

## Day 2 :

### 1. Pointers, Declare and Initialize pointers

A pointer is a variable whose value is the address of another variable.

Eg : int a =10;
int *ptr;
ptr=&a;

### 2. Function Definitions /invocations :

A function definition consists of function name, function parameters, return value and function's body. And calling a function is called function invocation.

### 3. Call functions by passing values and addresses:

**Call by value :** It is the method of passing arguments to a function, copies the actual value of an argument into the formal parameter of the function.

**Eg:** int a=10,b=20;

Swap(a,b);

**Call By Reference :** It is the method of passing arguments to a function, copies the address of an argument into the formal parameter.

**Eg :** swap(&a,&b);

### 4. Dynamic Memory Allocation /Deallocation:

**Allocation :** It is a memory management technique in which a program can request and return memory while it is executing.

Eg: new keyword is used in java to allocate the memory dynamically.

Hello obj=new Hello();

**Deallocation :** It is a process performed by a computer where a block of information is released from the memory so that it may be used by a different program.

The only way you can deallocate memory is by relying on the garbage collector. We can also call System.gc() to suggest that the garbage collector run immediately.

## 5. Memory Leakage :

It is a process in which a program persistently retains a computers primary memory. It occurs when the resident memory program does not release the allocated memory space even after execution.

## 6. User Defined Structures :

It's a data structure defined by the user, not the language. We need to implement them instead of using the implementation from the standard library.

## 7. Access Data Members:

Accessing the data member depends on the access control of that data member. If it's public, then the data member can be easily accessed using the direct member access (.) operator with the object of that class.

If, the data member is defined as private or protected, then we cannot access the data variables directly.Then we need to create special public member functions to access,use and initialize those members. These are also called getter and setter fuctions or Accessors and Mutator methods.

## 8. Dynamic memory allocation for user defined structures

It enables us to create structures of any size and length to suit our programs need within the program.

## 9. Array of pointers

Array of pointers is an indexed set of variables, where the variables are pointers referencing a location in memory.

## 10. Iterate on array of pointers

```
Eg: int a[]={1,2,3};
     int *arr[3];
    for(int i=0;i<3;i++){
         arr[i]=&a[i];
    }
   for(int i=0;i<3;i++){
         printf("%d\n",*arr[i]);
    }
```

## Day 3 :

### 1. Implement Data Structures :

A data structure is a particular way of organizing data in a computer so that it can be used effectively.

The implementation of a data structure usually requires writing a set of procedures that create and manipulate instances of that structure.

### 2. LinkedList

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers.
Methods: insertAtFirst(), insertAtLast(), insertAtMiddle(), delete() etc.
Applications : Image viewer, music player etc.

### 3. Stack

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO or FILO.

Methods : push(), pop(), display(), isEmpty(), peek().

Applications: Browser History, Expression evaluation, recursion etc.

## 4. Queue

A Queue is a linear structure which follows a particular order in which the operations are performed. The order is FIFO

Methods : enqueue(), dequeue(), display(), etc.

Applications: Serving the requests on a shared resource, call centre phone systems, etc.

## 5. Binary Tree

A tree represents the nodes connected by edges.
Binary Tree: A tree whose elements have at most 2 children.
Methods :
insert(item,root),delete(item,root),search(item,root),display(root),getChild(element,root),getParent(element,root) etc.
Applications : Store hierarchical data,XML/HTML data etc.