**Sunbeam Infotech**

**CPP Notes**

**Day 2**

**\*Escape Sequence**

-It is a character which is used to format the output.

- Following are the Escape Sequences

'\n' , '\t' , '\b' , '\r'

**\*Manipulator <iomanip>**

-It is a function which is used to format the output.

- endl, setw, setprecision, hex,dec,oct etc.

**Scope resolution operator:**

1) to call global functions

2) to define member functions of class outside the class

3 ) to access members of namespaces

**\* Access Specifier**

- If we want to control visibility of members of structure/class then we should use access specifier.

- defines the accessibility of data member,Member functions

Access specifiers in C++

1. private( - ) 2. protected( # ) 3. public( + )

1. Private - Can access inside the same struct/class in which it is declared Generally data members should declared as private. (data security)

2. public - Can access inside the same struct/class in which it is declared as well as inside out side function(like main()). Generally member functions should declared as public.

**Diffrence between class and struct in cpp is - all the members of class is by default private and all the members of struct in cpp is by default public.**

**\*Types of member function**

* **Constructor**
  + It is a member function of a class which is used to initialize object.
  + Due to following reasons, constructor is considered as special function of the class:
    - Its name is same as class name
    - It doesn't have any return type.
    - It is designed to call implicitly.
    - In the life time of the object is gets called only once
  + We can not declare constructor static, constant, volatile or virtual. We can declare constructor only inline.
  + Constructor overloading means inside a class more than one constructor is defined.
  + **Types of Constructor:**
    - Parameterless constructor
      * also called zero argument constructor or user defined default constructor
      * If we create object without passing argument then parameterless constructor gets called
    - Parameterized constructor
      * If we create object, by passing argument then paramterized constructor gets called
    - Default constructor
      * If we do not define constructor inside class then compiler generates default constructor for the class.
      * Compiler generated default constructor is parameterless.

* + **Constructor's member initializer list**
    - If we want to initialize data members according to users requirement then we should use constructor body.
    - Except array we can initialize any member inside constructors member initializer list.

* **Destructor**
  + It is a member function of a class which is used to release the resources.
  + It is considered as special function of the class
    - Its name is same as class name and always preceds with tild operator( ~ )
    - It doesnt have return type or doesn't take parameter.
    - It is designed to call implicitly.
  + Destructor calling sequence is exactly opposite of constructor calling sequence.
  + Destructor is designed to call implicitly but we can call it explicitly.
  + If we do not define destructor inside class then compiler generates default destructor for the class.
  + Default destructor do not deallocate resources allocated by the programmer. If we want to deallocate it then we should define destructor inside class.
* **Mutators/setter : modify state of object**
* **inspector/getter : do not change the state of the object**
* **facilitator**

**\*Modular Approach**

**\*Reference**

**-**Reference is derived data type.

-It alias or another name given to the exisiting memory location / object.

-It is mandatory to initialize reference.

-We can not create reference to constant value.

**\*Reference to array:**

**\*In C++, we can pass argument to the function using 3 ways:**

1. By Value

2. By Address

3. By Reference

**//We should not retrun non static local variable from function by address.**

**//If we want to return local variable from function by address then we should use static keyword.**

**//We should not return local variable from function by reference.**

**\* Returning object from a function / Copy constructor**