**Sunbeam Infotech**

**CPP Notes**

**Day 3**

**\* Dynamic Memory Allocation**

If we want to allocate memory dynamically then we should use new operator and to deallocate that memory we should use delete operator.

If pointer contains, address of deallocated memory then such pointer is called dangling pointer.

When we allocate space in memory, and if we loose pointer to reach to that memory then such wastage of memory is called memory leakage.

**\* Types of Copy**

* **Shallow Copy**
  + Process of copying state of object into another object as it is, is called shallow copy.
  + It is also called as bit-wise copy / bit by bit copy.
  + Following are the cases when shallow copy taken place:
    1. If we pass variable / object as a argument to the function by value.
    2. If we return object from function by value.
    3. If we initialize object:

Complex c2=c1

* + 1. If we assign the object

c2=c1;

* + 1. If we catch object by value.

Examples of shallow copy

Example 1: (Initialization)

int num1=50;

int num2=num1;

Example 2: (Assignment)

Complex c1(40,50);

c2=c1;

* **Deep Copy** 
  + It is also called as member-wise copy.
  + By modifing some state, if we create copy of the object then it is called deep copy.
* Steps to create deep copy
  + 1. Copy the required size from source object into destination object.
  + 2. Allocate new resource for the destination object.
  + 3. Copy the contents from resource of source object into destination object.

**\* Static Variable**

All the static and global variables get space only once during program loading / before starting execution of main function

* Static variable is also called as shared variable.
* Unintialized static and global variable get space on BSS segment.
* Intialized static and global variable get space on Data segment.
* Default value of static and global variable is zero.
* Static variables are same as global variables but it is having limited scope.

**\* Static Methods or Static Member Functions**

* Except main function, we can declare global function as well as member function static.
* To access non static members of the class, we should declare member function non static and to access
* static members of the class we should declare member function static.
* Member function of a class which is designed to call on object is called instance method. In short non static member function is also called as instance method.
* To access instance method either we should use object, pointer or reference to object.
* static member function is also called as class level method.
* To access class level method we should use classname and ::(scope resolution) operator.

**\*Constant**

* Constant in C++
  + const is type qualifier.
  + If we dont want modify value of the variable then we should use const keyword.
  + constant variable is also called as read only variable.
  + **We need to initialize const at the time of declaration in C++**.

**Constant Data Member**

* Once initialized, if we dont want to modify state of the data member inside any member function of the class including constructor body then we should declare data member constant.
* **If we declare data member constant then it is mandatory to initialize it using constructors member initializer list.**

**\* Mutable Keyword:**

In constant member function, if we want to modify state of non constant data member then we should use mutable keyword.

**\*Execption Handling**

If we give wrong input to the application then it generates runtime error/exception.

Exception is an object, which is used to send notification to the end user of the system if any exceptional situation occurs in the program.

If we want to manage OS resources carefully then we should use exception handling mechanism.

To handle exception then we should use 3 keywords:

1. try

* try is keyword in C++.
* If we want to inspect exception then we should put statements inside try block/handler.
* try block must have at least one catch block/handler

2. catch

* If we want to handle exception then we should use catch block/handler.
* Single try block may have multiple catch block.
* Catch block can handle exception thrown from try block only.
* A catch block, which can handle any type of exception is called generic catch block / catch-all handler.

3. throw

1. throw is keyword in C++.
2. If we want to generate exception explicitly then we should use throw keyword.
3. "throw statement" is a jump statement.

**\*Operator Overloading**

* operator is token in C/C++.
* It is used to generate expression.
* **operator is keyword** in C++.
* Types of operator:
  + Unary operator ( ++,--,&,!,~,sizeof())
  + Binary Operator (Arithmetic, relational, logical , bitwise, assignment)
  + Ternary operator (conditional)
* In C++, also we can not use operator with objects of user defined type directly.
* If we want to use operator with objects of user defined type then we should overload operator.
* To overload operator, we should define **operator function.**
* **We can define operator function using 2 ways:**
  + **Using member function**
  + **Using non member function**

**We can not overloading following operator using member as well as non member function:**

1. dot/member selection operator( . )
2. Pointer to member selectiion operator(.\*)
3. Scope resolution operator( :: )
4. Ternary/conditional operator( ? : )
5. sizeof() operator
6. typeid() operator
7. static\_cast operator
8. dynamic\_cast operator
9. const\_cast operator
10. reinterpret\_cast operator

**We can not overload following operators using non member function:**

1. Assignment operator( = )
2. Subscript / Index operator( [] )
3. Function Call operator[ () ]
4. Arrow / Dereferencing operator( -> )

**\*Operator overloading using member function( operator function must be member function )**

If we want to overload, binary operator using member function then **operator function should take only one parameter.**

c3 = c1 + c2; //will be called as ------ c3 = c1.operator+( c2 )

\* **Operator overloading using non member function(operator function must be global function)**

If we want to overload binary operator using non member function then **operator function should take two parameters.**

c3 = c1 + c2; //will be called as ------c3 = operator+(c1,c2);

**\* Template**

* If we want to write generic program in C++ then we should use template.
* Using template we can not reduce code size or execution time but we can reduce developers effort.
* It is designed for implementing generic data structure and algorithms.

Types of template:

1. Function Template 2. Class Template

**Syntax for Function Template**

**//template<typename T> //T : Type Parameter**

**\* Class Template**

In C++, by passing data type as a argument, we can write generic code hence parameterized type is called template