OOP (Week 02)

1 Constructor:

"A constructor is a member function that is automatically called when a class object is created."

- It is helpfull to think of constructor as initialization routine.
- Usefull for initializializing member variable.
- Have no return type. bcz constructor are not executed by explicit function calls and can not return a value.
- A constructor's purpose is to initialize an object's attributes. Bcz constructor execute as soon as object is created.

e.g:

```
classname(____){

}
```

(2) The Default Constructor:

"The constructor that takes no argument."

If we don't write the constructor in class, when class is compiled, C++ automatically write default constructor that does nothing.

(3) Destructor:

A destructor is a member function that is automatically called when an object is destroyed.

e.g:

```
~classname () {
}
```

→ Common use of destructor is to free/delete the memory when memory is dynamically allocated in presence of pointer

⇒ No return type
⇒ Can not accept any argument.

④ Destructor for pointer (ptr):

```
~classname () {
        delete ptr;
    }
```

⑤ Overloading Constructor:

"A class can have more than one constructor".

"Only 1 default constructor and 1 destructor."

⇒ When there are two constructors in a class. First one is default constructor and 2nd is Parameterized constructor In this way compiler would'nt be able to resolve which constructor to execute. known as overloading.

OOP (11/04/22)

COPY CONSTRUCTOR.

- Shallow copy Constructor

```
main() {
    {
        student sd;  } Scope of sd.
    } → wall.
}
```

- Student std2(sd); copy Constructor called

↳ If no copy constructor written by default shallow copy constructor written.

- Deep Copy Constructor (12/04/22)

```
Class MyClass{
        int *x;
    Public:
        MyClass(int i){
            x = new int;
            *x = i;
        }
        void printData(){
            cout<<*x; }
};
```

Assignment Operator:

```
    void operator = (const Student &obj)
    {      *rollNumber = *(obj.rollNumber);
    }
```

OOP **(18/07/22)**

```
Class Student{
    int a,b,c;
Public:
    Student(int i, int j, int k):a(i), b(j), c(k){
        // body of constructor
    }
};
```

Working

```
Student(int i, int j, int k): a(i*2), b(j+5), c(k+3){}
```

Error

```
Student(int i, int j, int k): b(j+5) a(i+b)
```

// Because a will initialize first (according to sequence of member variables) but value of b is not assigned.

Working

```
Student(int i, int j, int k): a(k), b(i), c(j){ }
```