CAP444
OBJECT ORIENTED PROGRAMMING USING C++

Created By:
Kumar Vishal
(SCA), LPU

Constructor and destructor

Constructors: types of constructors

- A special method which is used to initialize the object
- It is automatically called when an object of a class is created.
- it has the same name as the class name.
- it is always public
- it does not have any return type

Types of constructor:

- Default constructor
- Parameterized constructor
- Copy constructor

49:-
99:-
Bättre ljud
Bättre koll
Bättre skydd
Samla familjen, betala mindre.
599
SELFIEPINNE
99:-
Dubbel surf!
Dubbel surf!
299:-
299:-
Trasig telefon?

HTC
One
BlackBerry
Internet Devices
Verizon
Jetpack® 4G LTE
Mobile Hotspot
MiFi® 5510L
Touch to Explore
Choose a Device
Touch to Explore

Default constructor

A constructor which has no argument is known as default constructor. It is invoked at the time of creating object.

```
class Employee
 {
   public:
        Employee()
        {
            cout<<"Default Constructor"<<endl;
        }
};
```

Parameterized constructor

A constructor which has parameters is called parameterized constructor.

```
class Employee
 {
    int empId;
    string empName;
   public:
       Employee(int id, string name)
       {
          empId=id;
          empName=name;
       }
};
```

Copy Constructor

Copy Constructor is a type of constructor which is used to create a copy of an existing object of a class.

Syntax:

```
class-_name(class_name & object_name)
{
}
```

To call this: class_name obj1(arguments);

class_name obj2 = obj1;

Destructor

- Destructor is a special member function which destructs or deletes an object.
- A destructor is called automatically when object goes out of scope.
- Destructors have same name as the class preceded by a tilde (~)
- Destructor should not have any parameter
- There can only one destructor in a class
- When a class contains a pointer to memory allocated in class, we should write a destructor to release memory

Which option is correct for defining the destructor?

Option1:

```
~mobile()
      {
 cout<<"destructor called"<<endl;
      }
```

Option2:

```
 ~mobile( string str)
      {
 cout<<"destructor called"<<endl;
      }
```

A. Option1 is correct

B. Option2 is correct

C. Both option is correct

Any Query?