C++ Notes on Unit-III

Constructor, Destructor & Copy Constructor

By

Prof. P. J. Mokashi

Imp

* Construector in C++ *

In C++ a construetor is a special member function whose task is to initialize the objects of its class. It is special because its nam is same as the class name.

The construetor is invoke whereever an objects of its associate class is created. It is called construetor. because It constuets the values of data members of th specific class.

The general eepresentation of construetor inside every class is

```
Class College
{
   - - - -
   - - - -
public :
          College( )
          {
             - - - - -
             - - - - -
          }

   };
```

In the above eepresentation college is nothing but the name of the class and college ( ) is nothing but the name of the Constructor. Acceeding to the concept, the name of the constructor is same as class name. The Constructor have some special characteristics which are given below:

① Constructor should be decleared inside the public section of every class.

② They are invoke automatically when the objects are created.

③ They do not have any eeturntype not even void & therefore they cannot eeturn any value.

④ like functions, constructor have some default arguments.

⑤ An object with a constructor as well as ~~dest~~ destructor cannot be used as a member of union.

* ~~Parameterised~~ Constructor *

like functions we can place an arguments to the constructor. The arguments generally we can pass or placed inside the openning and closing brackets of every constructor. A constructor is said to be parameteries when it contains arguments in side it. A ~~constructor~~ is we can pass 'n' no of arguments to every constructor. similarly we can say that the constructor which contains 'n' no of parameters are called as parameteries constructor.

The general representation of parameteries constructor is shown below:

```
Class College
{
  - - - - -
  - - - - - -

public :
          College ( ) // constructor
          { - - - -      containing no
            - - - -      arguments.
          }
College (int a)
{ - - - -   // constructor containing
  - - - -      one arguments.
}

College( int a , int b)
{ - - - -   // constructor containing
  - - - -      two arguments.
}

};
```

In the above example college is the name of the class which is also declared as a constructor inside the public section of the class. The first constructor does not contains any arguments so it is called as constructor with no arguments. The second constructor will contain only one individual parameter so it is called as constructor with only one argument and the thired constructor contains two arguments so it is called as constructor with two arguments. Note that every constructor will either contains no arguments or it will contains 'n' no of arguments. Arguments are also called as parameters.

Imp
(4 marks) * multiple constructor inside a class
(constructor overloading)

like functions, we can place 'n' no of constructer with in the single class. such concept is secondly known as constructor overloading.

Defination :-
When a constructor shares a common name as class itself but they can differenciat with there parameter passing then the machanisum on the concept is known as constructor overloading.
The general representation of multiple constructor inside a single class is represented below:

```
Class college
{
  ~~Private~~      int a,b;

  Public :
           College ( )
           {
              cout<<" Constructor 1 " ;
           }
           College (inta)
           {
           cout<<" constructor 2 " ;
           }
           college (inta, int b)
           {
           cout<<" constructor 3 " ;
           }

};
           { Void main ()     College c1;
                            }
```

In the above example college is nothing but the name of the class which contains three constructors. All these three constructors are differenciate with there parameters passing.

So the example will fullfill the concept of constructor overloading i.e. multiple constructor inside a single class.

* Constructor with Default argument*

like functions, it is also possible to define or declare a constructors with some default values which are also called as default arguments. these arguments are nothing but some valid values pass to the constructors. The default arguments are place inside the openning and closing brackets of every constructor.

The general representation of the constructor with some default values are,

College (3, 4);
Sample (1.2, 1.5, 1.9);
where college and sample are two constructor.

The following program will demostrates the concept of constructor in C++.

```
// program for constructor
#include < iostream.h>
 class Integer
{ private:
            int m, n;

   public :
             Integer (int x, int y)
           {
               m=x ;
               n=y ;
           }
           void display ( )
           {
           cout<<"value of m ="<< m;
           cout<<" value of n ="<< n;
           }

     };
     void main ( )
    {
     Integer i1 (10,100);
     Integer i2 (50,71);
cout<<" Constructor 1:";
 i1.display();
cout<<" Constructor2:";
 i2.display();
}
```

The above program will generate an output as :

```
Constructor 1 :
value of m = 10
value of n = 100

Constructor 2 :
value of m = 50
value of n = 75
```

```
// Pgm for Constructor in C++
#include<iostream.h>
#include<conio.h>
class Integer
{
private: int m,n;
public: Integer(int x,int y)
{
m=x;
n=y;
}
void display()
{
cout<<"\n Value of m= "<<m;
cout<<"\n Value of n= "<<n;
}
};
void main( )
{
clrscr();
cout<<"\n\t\t ****** OUTPUT ******";
Integer i1(10,100);
Integer i2(30,250);
cout<<"\n Constructor 1:";
i1.display();
cout<<"\n Constructor 2:";
i2.display();
getch( );
}
```

```
                    ****** OUTPUT ******
 Constructor 1:
 Value of m= 10
 Value of n= 100
 Constructor 2:
 Value of m= 30
 Value of n= 250
```

★ Destructor :-

like constructor, destructor plays an importent eole in C++. Desstructor is basically used to destroy the objects or the memiey which has been created by a constructor. Like constructor, the destructor is also a member function whose name is same as class name but it is preceded with tilde sign (~).

eg:-

If we define a constructor with name sample then the destructor will eepresent with same name followed with ~~tilde~~ sign, the eepresentation is shown below:-

```
Class sample
{ -----
  ----

  Sample ( ) // constructor
  {
    ----
    ---
  }
~sample( ) // Destructor
{
  ----
  ----
}

};
```

The destructor never take any arguments not we can return any value. It will be invoke by the compilar then exit from the program. The destructor deallo--cate the memory space of or clean up the stoeage space which is no longer accessible.

Note that the destructor in a program is basically used to released the memory space. for farther used.

The following program will demonstrates the used of destructor in C++.

(used Alt + F5 key for destructor output

```
// program for Destructor
 Class Demo
{
  private :
            int a, b;

  Public :
            Demo( )
            {
            Cout<<" This is Constructor";
            }
           ~Demo( )
           {
           Cout<<" This is Destructor";
           }

};
void main( )
{
 Demo d1, d2 ;
}
```

The above program will generate an output as :

```
This is Constructor
This is Constructor
This is Destructor
This is Destructor.
```

```
// Pgm for Destructor
#include<iostream.h>
#include<conio.h>
class Demo
{
private: int a,b;
public: Demo()
{
cout<<"\n This is Constructor";
}
~  Demo()
{
cout<<"\n This is Destructor";
}
};
void main()
{
clrscr();
cout<<"\n\t\t\t ***** OUTPUT ******";
Demo d1,d2;
getch();
}
```

```
                        ***** OUTPUT ******
 This is Constructor
 This is Constructor
 This is Destructor
 This is Destructor
```

This is Destructor
This is Destructor

* Copy Constructor :- (Imp 6mark)
The copy constructor is also a member function which initializes an object using another object of the same class. Secondly we can also say ~~&~~ that a copy constructor is a constructor which creates an object by initializing it with an object of the same class which has been created previously.
A copy constructor is used to,
① Initializes one object from another of the same class.
② Copy an object to pass it as an argument to the function.
③ Copy an object to return it as an argument from the function.
If a copy constructor is not defined inside the class, the compilar it self defines one. if the class contains a pointer variable

Note that: a reference variable has been used as an argument to the copy constructor.

and has same dynamic memory allocations then in a such situation the class must contains. A copy constructor.

A copy constructor can be called

① When an object of a class is return by values.

② When an object of the class is passed to the function by value as an argument.

③ When an object is constructed based on another object of the same class.

④ When a compiler generates an temporary object.

The general syntax for representating copy constructor is :

```
Class name ( ~~Const~~ Class_name &obj )
{
    body of Constructor
}
```

```
classname (classname & obj)
{
    body of Constructor.
}
```

eg :-

```
Class Line
{
  -----
 pubic :
       Line(const Line &i1)
       {
         -----
       }
 };
```

As per the representation of a Copy constructor, it is basically used to declared and initialized an object from another object. The process of initializing an object through a copy constructor is known as <u>copy initialization</u>.

Note that a copy constructor can also used as a referance variable as its argument.

Following program will demonstates the used to copy constructor in C++.

```
// Pgm for Copy Constructor
#include<iostream.h>
#include<conio.h>
class Demo
{
private: int x,y;
public:
        Demo(int x1,int y1)
{
        x=x1;
        y=y1;
}
        Demo(const Demo &d2)
{
        x=d2.x;
        y=d2.y;
}
int getX()
{
        return x;
}
int getY()
{
        return y;
}     };
void main()
{
 clrscr();
 cout<<"\n\t\t\t ***** OUTPUT *****";
 Demo d1(10,15); //Normal Constructor is called
 Demo d2=d1;          // Copy Constructor is called
 cout<<"\n d1.x= "<<d1.getX();
 cout<<"\n d1.y= "<<d1.getY();
 cout<<"\n d2.x= "<<d2.getX();
 cout<<"\n d2.y= "<<d2.getY();
 getch();
}
```

```
                        ***** OUTPUT *****
 d1.x= 10
 d1.y= 15
 d2.x= 10
d2.y= 15
```

Question Bank on Unit- III

1) What is class? Explain how class can support the concept of data hiding?
2) Explain how class can be declared in C++ with example.
3) Write difference between Class in C++ & Structure in C.
4) What is object? Explain how class members can be accessed using objects with example.
5) Explain friend function with suitable program.
6) What is constructor? Explain with suitable program.
7) Explain how multiple constructors can be declared inside a class with example.
8) What is destructor? Explain with suitable program.
9) Write difference between Constructor & destructor.
10) Explain copy constructor with suitable program.
11) Write difference between Constructor and Copy constructor.
12) Write a program which demonstrates the concept of class & objects in C++.