SDF II(15B11CI211)

EVEN Semester 2021

2nd Semester , First Year

Jaypee Institute Of Information Technology (JIIT), Noida

Lecture 7 – Copy Constructor

Copy constructor

- The copy constructor is a constructor which creates an object by initializing it with an object of the same class, which has been created previously.

Syntax for copy constructor

- ClassName (const ClassName &old_obj);

Copy Constructor

```
#include<iostream>
using namespace std;
  class Point
{
private:
    int x, y;
 public:
    Point(int x1, int y1)
    { x = x1;
     y = y1; }
      // Copy constructor
   Point(const Point &p2)
{x = p2.x;
 y = p2.y; }
    int getX()
  {  return x;
}
    int getY()
    {  return y;
}
};
```

```
int main()
{
    Point p1(10, 15);     // Normal constructor is called here
    Point p2 = p1;                // Copy constructor is called here

    // Let us access values assigned by constructors
    cout << "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();
    cout << "\np2.x = " << p2.getX() << ", p2.y = " << p2.getY();

    return 0;
}
```

Output

p1.x = 10, p1.y = 15
p2.x = 10, p2.y = 15

Type of Copy Constructor

- Default
- User defined

Two types of copies are produced by the constructor:

- Shallow copy
- Deep copy

Shallow copy

- The default copy constructor can only produce the shallow copy.
- A Shallow copy is defined as the process of creating the copy of an object by copying data of all the member variables as it is.

```
include <iostream>

using namespace std;

class Demo
{
   int a;
   int b;
   int *p;
   public:
   Demo()
   {
      p=new int;
   }
void setdata(int x,int y,int z)
   {
      a=x;
      b=y;
      *p=z;
   }
void showdata()
   {
      std::cout << "value of a is : " <<a<< std::endl;
      std::cout << "value of b is : " <<b<< std::endl;
      std::cout << "value of *p is : " <<*p<< std::endl;
   }
};
int main()
{
  Demo d1;
  d1.setdata(4,5,7);
  Demo d2 = d1;
  d2.showdata();
    return 0;
}
```

Output

```
value of a is : 4
value of b is : 5
value of *p is : 7
```

Deep copy

- Deep copy dynamically allocates the memory for the copy and then copies the actual value, both the source and copy have distinct memory locations.

```
#include <iostream>
using namespace std;
class Demo
{
   public:
   int a;
   int b;
   int *p;

   Demo()
   {
      p=new int;
   }
   Demo(Demo &d)
   {
      a = d.a;
      b = d.b;
      p = new int;
      *p = *(d.p);
   }
   void setdata(int x,int y,int z)
   {
      a=x;
      b=y;
      *p=z;
   }

void showdata()
   {
      std::cout << "value of a is : " <<a<< std::endl;
      std::cout << "value of b is : " <<b<< std::endl;
      std::cout << "value of *p is : " <<*p<< std::endl;
   }
};
int main()
{
  Demo d1;
  d1.setdata(4,5,7);
  Demo d2 = d1;
  d2.showdata();
  return 0;
}
```

Output

```
value of a is : 4
value of b is : 5
value of *p is : 7
```

Copy constructor vs Assignment Operator

- MyClass t1, t2;
- MyClass t3 = t1; // ----> (1) copy constructor
- t2 = t1; // -----> (2) assignement operator

- Copy constructor is called when a new object is created from an existing object, as a copy of the existing object.
- Assignment operator is called when an already initialized object is assigned a new value from another existing object.

Object Initialization

```
void main()
{
    Bank b1;                    // creating object through default constructor
Bank *b2=new Bank()             // Another way of creating object through default constructor
Bank b3=b1;                     // creating object with copy constructor
Bank b4(b1);                    // Another way of creating object with copy constructor
Bank *b5=new Bank(b1)                   // Another way of creating object with copy constructor
 Bank b6(0.0,0.0);                      // creating object with parameterized constructor
```

- An object is a instance of a class. Resources are allocated when an object is initialized.

Accessing members through objects

```
void main()
{
 Bank b1;
Bank *b2=new Bank()
B1.acc_no=121;
b1.deposit();
b2->deposit();
b1. check_balance();
}
```

References

- https://www.geeksforgeeks.org/copy-constructor-in-cpp/
- https://www.javatpoint.com/cpp-copy-constructor