27/8/2021

Name: Durvesh Naresh Patil
PRN : 2019BTEEN00035

Object Oriented Programming

Exp. 4 : Constructor and Destructor.

⊙ Activity :

1) Write a class 'Book' without constructor & check the results. Write the code and result.

⇒

```
# include <iostream>
using namespace std;

class Book
{
  private :
      int id ;
      string name;
      int price;
  public :
      void setData ()
      {
          cout << "\n Enter book id: ";
          cin >> id ;
          cout << "\n Enter book name: ";
          cin >> name;
          cout << "\n Enter book price: ";
          cin >> price ;
      }
      void display ()
      {
          cout << "\n Book Data \n Id : " << id << "\n Name: "
              << name << "\n Price : " << price ;
      }
```

```
    int getId()
    {
        return id;
    }
    int getPrice()
    {
        return price;
    }
};

int main()
{
    Book b1;
    b1.setData();
    b1.display();
    return 0;
}
```

<u>Input</u> : Id : 121
Name : Rich Dad Poor Dad
Price : 200

<u>Output</u> :
Book Data
Id : 121
Name : Rich Dad Poor Dad
Price : 200

2) Write one constructor in same class & now check the results. Write code and result.

⇒

```
# include <iostream>
using namespace std;

class Book
{
    private :
        int id ; string name ; int price ;
    public :
        Book ()
        {
            cout <<"\n Constructor is called" ;
        }
        ~Book ()
        {
            cout <<" \n Destructor is called" ;
        }
        void setData ()
        {
            cout <<"\n Enter id : " ;  cin >> id ;
            cout << "\n Enter name : " ; cin >> name ;
            cout <<"\n Enter price : " ; cin >> price ;
        }
        void display ()
        {
            cout << "\n Book Data \n Id : "<< id << "\n Name : "
                << name << "\n Price : "<< price ;
        }
```

```
    int getId()
    {
        return id;
    }
    int getPrice()
    {
        return price;
    }
};

int main()
{
    Book b1;
    b1.setData();
    b1.display();
    return 0;
}
```

<u>Input</u> : Enter Id : 121
Enter Name : Rich Dad Poor Dad
Enter price : 200

<u>Output</u> : Constructor is called
Book Data
Id : 121
Name : Rich Dad Poor Dad
Price : 200
Destructor is called

Add one more constructor in same class and
3) show initializing using constructor.
⇒

```
# include <iostream>
using namespace std;

class Book
{
  private:
      int id; string name; int price;
  public:
      Book()
      {
         cout << "\n Constructor is called";
      }
      Book( int id, string name, int price)
      {
          this.id = id;
          this.name = name;
          this.price = price;
      }
      ~ Book()
      {
          cout << "\n Destructor is called";
      }
      void setData()
      {
          cout << "\n Enter id: "; cin >> id;
          cout << "\n Enter name: "; cin >> name;
          cout << "\n Enter price: "; cin >> price;
      }
```

```
    void display()
    {
        cout << "\n Book Data\n Id: " << id << "\n Name: "
            << name << "\n Price: " << price;
    }
};

int main()
{
    Book b1(121, "Rich Dad Poor Dad", 200);
    b1.display();

    return 0;
}
```

<u>Output</u> :

```
Book Data
Id: 121
Name: Rich Dad Poor Dad
Price: 200
Destructor is called.
```

Questions :

1) What is the role of a constructor in classes ?
⇒ To initialize the data members of an object when it is created.

2) What is a copy constructor ?
⇒ A constructor to initialize an object with the values of another object.

3) What happens if a user forgets to define constructor inside a class ?
⇒ Compiler provides a default constructor to avoid errors.

4) How constructors are different from other member functions of the class ?
⇒ All of the mentioned. (a) Has the same name as the class itself. b) Constructor do not return anything. c) Constructors are automatically called when an object is created.

5) When destructors are called ?
⇒ All of the mentioned. (a) when a program ends. b) When a function ends. c) when a delete operator is used).

⊙ <u>Conclusion</u> : Constructor & destructor don't have returning type not even void.

* <u>Constructors</u> : (i) Constructors are used to initialize object.
(ii) Constructors can be overloaded.
(iii) Constructor function getting call automatically at time of declaring object.
(iv) Default, parameterised & copy are the three types of constructors.

* <u>Destructors</u> : (i) Destructors are used to release external resources associated with object.
(ii) Destructors cannot be overloaded.
(iii) Destructors cannot called explicitly, these are getting called automatically when program execution is out of scope where object created.