SRM UNIVERSITY AP Amaravati

Object Oriented Programming using C++

CSE-III

CSE206-OOP

Dr. Priyanka Singh

Derived Class Constructor

Derived Class Constructor

- If the Base class has only the default constructor, no need of explicit constructor in Derived Class.

```
#include <iostream>
using namespace std;
class Base{
  int a;
  public:
    Base()
    { a=5;}
    void show(){
        cout<<"Base class Constructor Called, Value of a is: "<<a;
    }
};
class Derived: public Base{
};
int main()
{
    Derived d;
    d.show();
    return 0;
}
```

Derived Class Constructor

If the Base class has parametrized constructor, it is mandatory for the Derived class to have a constructor and pass the arguments to the Base class constructor.

```
class Base{
  int a;
  public:
    Base(int arg){       a=arg;   }
    void show(){
        cout<<"Base class Constructor Called, Value of a is: "<<a;
    }
};
class Derived: public Base
{
    public:
        Derived(int x):Base(x)
{ }
};
int main()
{
    Derived d(6);
    d.show();
    return 0;
}
```

Derived Class Constructor

Whenever an object of derived class is created, the Base class constructor is executed first and then the derived class constructor.

```
#include <iostream>
using namespace std;
class Base{
  public:
    Base(){
        cout<<"Base class Constructor Called"<<endl;
    }
};
class Derived: public Base{
    public:
        Derived(){
            cout<<"Derived class Constructor Called"<<endl;
        }
};
int main()
{
    Derived d;
    return 0;
}
```

Hands On (Inheritance, Constructor, Method Overriding)

```
#include<iostream>
using namespace std;
class Base1 {
public:   Base1()
          { cout << " Base1's constructor called" << endl; }
};
class Base2 {public:
          Base2()
          { cout << "Base2's constructor called" << endl; }
};
class Derived: public Base1, public Base2 {
public:   Derived()
          { cout << "Derived's constructor called" << endl; }
};
int main(){
Derived d;
return 0;
}
```

```
Base1′s constructor called
Base2′s constructor called
Derived’s constructor called
```

Hands On (Inheritance, Constructor, Method Overriding)

```
#include<iostream>
using namespace std;
class Base {
public:
        void fun()          { cout << "Base::fun() called";}
        void fun(int i)      { cout << "Base::fun(int i) called"; }
};
class Derived: public Base {
public:
        void fun() {         cout << "Derived::fun() called"; }
};
int main() {
        Derived d;
        d.Base::fun(5);
        return 0;
}
```

```
Base::fun(int i) called
```

Hands On (Inheritance, Constructor, Method Overriding)

```
#include<iostream>
using namespace std;
class Base
{public:
        int fun() { cout << "Base::fun() called"; }
        int fun(int i) { cout << "Base::fun(int i) called"; }
};
class Derived: public Base{
public:
        int fun() { cout << "Derived::fun() called"; }
};
int main()
{       Derived d;
        d.fun(5);
        return 0;
}
```

Compiler Error

Hands On (Inheritance, Constructor, Method Overriding)

```
#include<iostream>
using namespace std;
class P {
public:
void print() { cout <<" Inside P"; }
};
class Q : public P {
public:
void print() { cout <<" Inside Q"; }
};
class R: public Q { };
int main(void)
{R r;
r.print();
return 0;
}
```

```
Inside Q
```

Practice question-1

We want to calculate the total marks of each student of a class in Physics, Chemistry and Mathematics and the average marks of the class. The number of students in the class are entered by the user. Create a class named Marks with data members for roll number, name and marks. Create three other classes inheriting the Marks class, namely Physics, Chemistry and Mathematics, which are used to define marks in individual subject of each student. Roll number of each student will be generated automatically.

Tutorial

Tutorial question - 1

We want to store the information of different vehicles. Create a class named Vehicle with two data member named mileage and price. Create its two subclasses

***Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol).**

***Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches)**

Make another two subclasses Audi and Ford of Car, each having a data member to store the model type. Next, make two subclasses Bajaj and TVS, each having a data member to store the make-type.

Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.) Do the same for a Bajaj and a TVS bike.

Tutorial question - 2

Create a class named Shape with a function that prints "This is a shape". Create another class named Polygon inheriting the Shape class with the same function that prints "Polygon is a shape". Create two other classes named Rectangle and Triangle having the same function which prints "Rectangle is a polygon" and "Triangle is a polygon" respectively. Again, make another class named Square having the same function which prints "Square is a rectangle". Now, try calling the function by the object of each of these classes.

Tutorial question - 3

- All the banks operating in India are controlled by RBI. RBI has set a well defined guideline (e.g. minimum interest rate, minimum balance allowed, maximum withdrawal limit etc) which all banks must follow. For example, suppose RBI has set minimum interest rate applicable to a saving bank account to be 4% annually; however, banks are free to use 4% interest rate or to set any rates above it.

Write a program to implement bank functionality in the above scenario. Note: Create few classes namely Customer, Account, RBI (Base Class) and few derived classes (SBI, ICICI, PNB etc). Assume and implement required member variables and functions in each class.