```
    BaseClass2()
    { cout << "BaseClass2 constructor called" << endl;  }
};

class DerivedClass: public BaseClass1, public BaseClass2 {
  public:
   DerivedClass()
    {  cout << "DerivedClass constructor called" << endl;  }
};

int main()
{
  DerivedClass derived_class;
  return 0;
}
```

Output:

```
BaseClass1 constructor called
BaseClass2 constructor called
DerivedClass constructor called
```

Reason:
The above program demonstrates Multiple inheritances. So when the Derived class's constructor is called, it automatically calls the Base class's constructors from left to right order of inheritance.

2. What will be the output of the below code?

```
class Scaler
{
   static int i;

   static
   {
       System.out.println("a");

       i = 100;
   }
}

public class StaticBlock
{
   static
```