OBJECT ORIENTED PROGRAMMING USING JAVA

Example 1

```
public class First_C {
     public void myMethod()
    {
    System.out.println("Method");
    }
     public void First_C()
    {
    System.out.println("Constructor ");
    }
    static {
        System.out.println("static block");  }
    public static void main(String[] args) {
    First_C c = new First_C();
    c.First_C();
    c.myMethod();  }  }
```

OUTPUT:
static block
Constructor
Method

Example 2

```
class Parent {
    public void Print()
    {
        System.out.println("Parent");
    }}
class Child extends Parent {
    public void Print()
    {
        System.out.println("Child");
    }}
class Main {
    public static void PrintMain(Parent o)
    { o.Print();   }
    public static void main(String[] args)
    {
        Parent x = new Parent();
        Parent y = new Child();
        Child z = new Child();
        PrintMain(x);  PrintMain(y);  PrintMain(z); }}
```

OUTPUT:
Parent
Child
Child

Example 3

```
class CodeA
{  public String type = "A ";
   public CodeA() {
      System.out.print("CodeA ");
   }}
public class CodeB extends CodeA
{
   public CodeB() {
      System.out.print("CodeB ");}
   void go()
   {  type = "B ";
      System.out.print(this.type + super.type);
   }
   public static void main(String[] args)
   {
      new CodeB().go();
   }}
```

OUTPUT:
CodeA CodeB B A

Example 4

```
class Test1 {
 Test1(int x)
    {
 System.out.println("Constructor called " + x);
    }
}
class Test2 {
Test2(int i) { t1 = new Test1(i); }
 public static void main(String[] args)
    {
      Test2 t2 = new Test2(5);
    }
}
```

OUTPUT:
Constructor called 5

Example 5

```
class Base {
    public String name = "Base";
}
class Derived extends Base {
    private String name = "Derived";
    public String getName() {
        return name;
    }
}
public class Details {
    public static void main(String[] args) {
        System.out.println(new Derived().getName());
        System.out.println(new Derived().name);
    }
}
```

OUTPUT:
Compilation error

Problem Statement 1

Write a Java class Author with following features:

Instance variables :

firstName for the author's first name of type String.

lastName for the author's last name of type String.

Constructor:

public Author (String firstName, String lastName): A constructor with parameters, it creates the Author object by setting the two fields to the passed values.

Instance methods:

public void setFirstName (String firstName): Used to set the first name of author.

public void setLastName (String lastName): Used to set the last name of author.

public String toString(): This method printed out author's name to the screen

THANK YOU