Practical 04: Encapsulation & Inheritance

Exercise 01:

Create a class called “Employee” which has 3 private variables (empID, empName, empDesignation) and create getters and setters for each field. Please note that this has no main method since this is just a blueprint not a application. Now crate a test class to invoke the Employee class. Create two objects for Mr.Bogdan and Ms.Bird and set required values using setters and print them back on the console using getters.

```
public class Employee

{

   //data

   private int empID;

   private String empName;

   private String empDesignation;


   //setter method to empID

   public void setId(int id)

   {

      empID=id;

   }

   //getter method to empID

   public int getId()

   {

      return empID;

   }

   //setter method to empName

   public void setName(String name)

   {

      empName=name;
```

```
    }

    //getter method to empName

    public String getName()

    {

        return empName;

    }

    //setter method to empDesignation

    public void setDesignation(String d)

    {

        empDesignation=d;

    }

    //getter method to empDesignation

    public String getDesignation()

    {

        return empDesignation;

    }

}


public class Test

{

    public static void main(String[] args)

    {

        Employee e1=new Employee();

        e1.setId(1234);

        e1.setName("Mr.Bogdan");
        e1.setDesignation("HR Manager");

        System.out.println("Employee id: "+e1.getId());

        System.out.println("Employee Name: "+e1.getName());

        System.out.println("Employee Designation: "+e1.getDesignation());


        System.out.print("\n");

        Employee e2=new Employee();

        e2.setId(1256);

        e2.setName("Ms.Bird");

        e2.setDesignation("Supervisor");

        System.out.println("Employee id: "+e2.getId());

        System.out.println("Employee Name: "+e2.getName());

        System.out.println("Employee Designation: "+e2.getDesignation());

    }

}
```

Exercise 02:

Develop the following class execute and discuss the answer: Please note that each class stored in separate files. Write down the answer.

```
class SuperB {

   int x;

   void setIt (int n) { x=n;}

   void increase () { x=x+1;}

   void triple () {x=x*3;};

   int returnIt () {return x;}

}
```

```
class SubC extends SuperB {

   void triple () {x=x+3;} // override existing method

   void quadruple () {x=x*4;} // new method

}

public class TestInheritance {

   public static void main(String[] args) {

      SuperB b = new SuperB();

      b.setIt(2);

      b.increase();

      b.triple();

      System.out.println( b.returnIt() );

      SubC c = new SubC();

      c.setIt(2);

      c.increase();

      c.triple();

      System.out.println( c.returnIt() ); }

}
```

Output:

```
9

6
```