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 {
   private int empID;
   private String empName;
   private String empDesignation;

   public int getEmpID() {
      return empID;
   }

   public void setEmpID(int empID) {
      this.empID = empID;
   }

   public String getEmpName() {
      return empName;
   }

   public void setEmpName(String empName) {
      this.empName = empName;
```

```
    }

    public String getEmpDesignation() {
        return empDesignation;
    }

    public void setEmpDesignation(String empDesignation) {
        this.empDesignation = empDesignation;
    }
}

class Main {
    public static void main(String[] args) {
        Employee e1 = new Employee();
        e1.setEmpID(1);
        e1.setEmpName("John Doe");
        e1.setEmpDesignation("Software Engineer");

        System.out.println("Employee ID: " + e1.getEmpID());
        System.out.println("Employee Name: " + e1.getEmpName());
        System.out.println("Employee Designation: " + e1.getEmpDesignation());
    }
}
```

Practical 04: Encapsulation & Inheritance

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() ); }
```

```
}
```

print out : 9 6