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;

//getter
public String getEmpID(){
return empID;
}
//setter
Public void setEmpID(int empID){
this.empID=newEmpID;
}

//getter
public String getEmpName(){
return empName;
}
//setter
Public void setEmpName(String empID){
this.empName=newEmpName;
}

//getter
public String getEmpDesignation(){
return empDesignation;
}
//setter
Public void setEmpDesignation(String empDesignation){
this.empDesignation=newEmpDesignation;
}
}
```

```
Public class test{
public static void main(String[] args) {

Employee Employee1 = new Employee();
Employee1.setEmpID(2001);
Employee1.setEmpName(“Mr.Bodegn”);
Employee1.setEmpDesignation(“Manager”);

Employee Employee2=new Employee();
Employee2.setEmpID(2002);
Employee2.setEmpName(“Ms.bird”);
Employee2.setEmpDesignation(“Secrty”);

System.out.println(“Employee ID”+Employee1.getEmpID());
System.out.println(“Employee
Name”+Employee1.getEmpName());
System.out.println(“Employee
Designation”+Employee1.getEmpDesignation());

System.out.println(“Employee ID”+Employee2.getEmpID());
System.out.println(“Employee
Name”+Employee2.getEmpName());
System.out.println(“Employee
Designation”+Employee2.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() );
 // output 9
    SubC c = new SubC();
    c.setIt(2);
    c.increase();
    c.triple();
    System.out.println( c.returnIt() );
// output 6
}
}
```