Practical 04 – Encapsulation & Inheritance

Exercise 01:

```
package com.mycompany.employeetest;

public class EmployeeTest
{
   public static void main(String[] args)
   {
      Employee mrBogdon=new Employee();
      Employee mrBird=new Employee();


      mrBogdon.setEmpID(1001);
mrBogdon.setEmpName("Mr.Bogdon");
mrBogdon.setEmpDesignation("Web Developer");


      mrBird.setEmpID(1002);
mrBird.setEmpName("Mr.Bird");
      mrBird.setEmpDesignation("Fullstack Developer");


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


      System.out.println("\nEmployee ID: "+mrBird.getEmpID());
System.out.println("Employee Name: "+mrBird.getEmpName());
```

```
        System.out.println("Employee Designation: "+mrBird.getEmpDesignation());
    }
}
package com.mycompany.employeetest;
public class Empolyee
{
    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.empID=empID;
```

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

Exercise 02:

Output :

9 6

Class SuperB

- This class represents a superclass containing four methods: setIt, increase, triple, and returnIt.
- The instance variable x's value is set with the setIt method.
- The increase method increases the value of x by 1.
- The value of x is multiplied by three using the triple method. The returnIt method returns the current value of x.

class SubC extends SuperB

- In order to become a subclass of SuperB, this class extends the SuperB class.
- Instead of multiplying x by 3, it adds 3 to the x value instead of using the triple function from the superclass.

- Additionally, it adds a new method called quadruple, which multiplies the value of x by 4.

public class TestInheritance

- This class has the main method where the behavior of inheritance is tested.
- In the beginning, it creates a SuperB instance called b.
- Using the setIt method, it initializes b to 2 in this case.
- Following that, it increases b's value by 1, making x equal to 3.
- The triple method is then used to triple the value of b, so x becomes 9.
- Finally, using the returnIt method, it prints the current value of b. then it makes a SubC instance with the name c.
- Using the setIt method, it initializes c to the value 2.
- Then it uses the increase method to increase the C value by 1, so x becomes .
- Then, using the triple method, which is overridden in the SubC class, it triples the value of c, so x becomes 6.
- Using the returned method, it prints the current value of C.

Exercise 03:

```
package com.mycompany.testclass;

public class Person

{

   private String name;

private int id;

   public Person(String name, int id)

   {

      this.name=name;

this.id=id;

   }
```

```
   public String getName()
   {
      return name;
   }
   public int getID()
   {
      return id;
   }
   public void setName(String name)
   {
      this.name=name;
   }
   public void setID(int id)
   {
this.id=id;
   }
}
package com.mycompany.testclass;
public class Student extends Person
{
   private String course;
   public Student(String name,int id,String course)
   {
```

```
        super(name,id);
this.course=course;
    }
    public String getCourse()
    {
        return course;
    }
    public void setCourse(String course)
    {
      this.course=course;
    }
}
package com.mycompany.testclass;
public class Lecturer extends Person
{
    private String programme;
    public Lecturer(String name,int id,String programme)
    {
        super(name,id);
this.programme=programme;
    }
    public String getProgramme()
    {
        return programme;
```

```
    }
    public void setProgramme(String programme)
    {
        this.programme=programme;
    }
}
package com.mycompany.testclass;
public class TestClass
{
    public static void main(String[] args)
    {
        Student student=new Student("Anne Watson", 1002, "Data Science");
        System.out.println("Student Name: "+student.getName());
        System.out.println("Student ID: "+student.getID());
        System.out.println("Student Course: "+student.getCourse());
        System.out.println("");
        Lecturer lecturer=new Lecturer("Tom Daniel",5440,"Computer
Architecture");
        System.out.println("Lecturer Name: "+lecturer.getName());
        System.out.println("Lecturer ID: "+lecturer.getID());
        System.out.println("Lecturer Programme: "+lecturer.getProgramme());
    }
}
```

Exercise 04:

Output:

true true

true