**JAVA LAB SHEET 04**

**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 and Setter for empID

public int getEmpID() {

return empID;

}

public void setEmpID(int empID) {

this.empID = empID;

}

// Getter and Setter for empName

public String getEmpName() {

return empName;

}

public void setEmpName(String empName) {

this.empName = empName;

}

// Getter and Setter for empDesignation

public String getEmpDesignation() {

return empDesignation;

}

public void setEmpDesignation(String empDesignation) {

this.empDesignation = empDesignation;

}

}

public class TestEmployee {

public static void main(String[] args) {

// Creating Employee objects

Employee mrBogdan = new Employee();

Employee msBird = new Employee();

// Setting values for Mr. Bogdan

mrBogdan.setEmpID(101);

mrBogdan.setEmpName("Bogdan");

mrBogdan.setEmpDesignation("Software Engineer");

// Setting values for Ms. Bird

msBird.setEmpID(102);

msBird.setEmpName("Bird");

msBird.setEmpDesignation("Project Manager");

// Printing values using getters

System.out.println("Mr. Bogdan's Details:");

System.out.println("ID: " + mrBogdan.getEmpID());

System.out.println("Name: " + mrBogdan.getEmpName());

System.out.println("Designation: " + mrBogdan.getEmpDesignation());

System.out.println();

System.out.println("Ms. Bird's Details:");

System.out.println("ID: " + msBird.getEmpID());

System.out.println("Name: " + msBird.getEmpName());

System.out.println("Designation: " + msBird.getEmpDesignation());

}

}

Mr. Bogdan's Details:

ID: 101

Name: Bogdan

Designation: Software Engineer

Ms. Bird's Details:

ID: 102

Name: Bird

Designation: Project Manager

**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() ); }**

**}**

**SuperB** class:

class SuperB {

int x;

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

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

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

int returnIt() { return x; }

}

**SubC** class:

class SubC extends SuperB {

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

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

}

**TestInheritance** class (main 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

**Exercise 03:**

**Recall the following scenario discussed during the class. Develop a code base to represent the scenario. Add a test class to invoke Lecturer and Student class by creating atleast one object from each.**

**Note: All the common attributes and behavior stored in the super class and only the specific fields and behavior stored in subclasses.**

|  |
| --- |
| **Student** |
| * **name** |
| * **id** |
| * **course** |
| **+ setName()/getName()** |
| **+ setID()/getID()** |
| **+ setCourse()/getCourse()** |

|  |
| --- |
| **Lecturer** |
| * **name** |
| * **id** |
| * **programme** |
| **+ setName()/getName()** |
| **+ setID()/getID()** |
| **+ setProg()/getProg()** |

|  |
| --- |
| **Person** |
| **Identify field and attributes to be stored in this class** |

**Person** class:

public class Person {

private String name;

private int id;

// Getter and Setter for name

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

// Getter and Setter for id

public int getID() {

return id;

}

public void setID(int id) {

this.id = id;

}

}

**Student** class:

public class Student extends Person {

private String course;

// Getter and Setter for course

public String getCourse() {

return course;

}

public void setCourse(String course) {

this.course = course;

}

}

**Lecturer** class:

public class Lecturer extends Person {

private String programme;

// Getter and Setter for programme

public String getProg() {

return programme;

}

public void setProg(String programme) {

this.programme = programme;

}

}

1. **TestPerson** class (to test **Student** and **Lecturer** classes):

public class TestPerson {

public static void main(String[] args) {

// Creating a Student object

Student student = new Student();

student.setName("John Doe");

student.setID(101);

student.setCourse("Computer Science");

// Creating a Lecturer object

Lecturer lecturer = new Lecturer();

lecturer.setName("Dr. Jane Smith");

lecturer.setID(201);

lecturer.setProg("Computer Engineering");

// Printing details of the Student and Lecturer

System.out.println("Student Details:");

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

System.out.println("ID: " + student.getID());

System.out.println("Course: " + student.getCourse());

System.out.println();

System.out.println("Lecturer Details:");

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

System.out.println("ID: " + lecturer.getID());

System.out.println("Programme: " + lecturer.getProg());

}

}

output:

Student Details:

Name: John Doe

ID: 101

Course: Computer Science

Lecturer Details:

Name: Dr. Jane Smith

ID: 201

Programme: Computer Engineering

**Exercise 04**

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

**public class Animal{}**

**public class Mammal extends Animal{}**

**public class Reptile extends Animal{}**

**public class Dog extends Mammal{**

**public static void main(String args[]){**

**Animal a = new Animal();**

**Mammal m = new Mammal();**

**Dog d = new Dog();**

**System.out.println(m instanceof Animal);**

**System.out.println(d instanceof Mammal);**

**System.out.println(d instanceof Animal);**

**}**

**}**

**Animal** class:

public class Animal {}

**Mammal** class:

public class Mammal extends Animal {}

**Reptile** class:

public class Reptile extends Animal {}

**Dog** class:

public class Dog extends Mammal {

public static void main(String args[]) {

Animal a = new Animal();

Mammal m = new Mammal();

Dog d = new Dog();

System.out.println(m instanceof Animal);

System.out.println(d instanceof Mammal);

System.out.println(d instanceof Animal);

}

}

Output

true

true

true