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;**

**}**

**}**

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**

**8**

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 (Superclass)**

**public class Person {**

**private String name;**

**private int id;**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public int getId() {**

**return id;**

**}**

**public void setId(int id) {**

**this.id = id;**

**}**

**}**

**// Student class (Subclass)**

**public class Student extends Person {**

**private String course;**

**public String getCourse() {**

**return course;**

**}**

**public void setCourse(String course) {**

**this.course = course;**

**}**

**}**

**// Lecturer class (Subclass)**

**public class Lecturer extends Person {**

**private String programme;**

**public String getProgramme() {**

**return programme;**

**}**

**public void setProgramme(String programme) {**

**this.programme = programme;**

**}**

**}**

**public class TestPerson {**

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

**Student student = new Student();**

**student.setName("John");**

**student.setId(101);**

**student.setCourse("Computer Science");**

**Lecturer lecturer = new Lecturer();**

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

**lecturer.setId(201);**

**lecturer.setProgramme("Data Science");**

**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.getProgramme());**

**}**

**}**

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);

}

}

**Output**

**true**

**true**

**true**