1.public class Employee {

private int empID;

private String empName;

private String empDesignation;

// Getter for empID

public int getEmpID() {

return empID;

}

// Setter for empID

public void setEmpID(int empID) {

this.empID = empID;

}

// Getter for empName

public String getEmpName() {

return empName;

}

// Setter for empName

public void setEmpName(String empName) {

this.empName = empName;

}

// Getter for empDesignation

public String getEmpDesignation() {

return empDesignation;

}

// Setter for empDesignation

public void setEmpDesignation(String empDesignation) {

this.empDesignation = empDesignation;

}

}

public class TestEmployee {

public static void main(String[] args) {

// Create two Employee objects

Employee mrBogdan = new Employee();

Employee msBird = new Employee();

// Set values using setters

mrBogdan.setEmpID(1001);

mrBogdan.setEmpName("Mr. Bogdan");

mrBogdan.setEmpDesignation("Software Engineer");

msBird.setEmpID(1002);

msBird.setEmpName("Ms. Bird");

msBird.setEmpDesignation("Product Manager");

// Print the details using getters

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

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

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

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

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

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

}

}

2. 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();

3. public class Person {

private String name;

private int id;

// Constructor

public Person(String name, int id) {

this.name = name;

this.id = id;

}

// Getters and setters

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;

}

}

public class Student extends Person {

private String course;

// Constructor

public Student(String name, int id, String course) {

super(name, id);

this.course = course;

}

// Getter and setter for course

public String getCourse() {

return course;

}

public void setCourse(String course) {

this.course = course;

}

}

public class Lecturer extends Person {

private String programme;

// Constructor

public Lecturer(String name, int id, String programme) {

super(name, id);

this.programme = programme;

}

// Getter and setter for programme

public String getProgramme() {

return programme;

}

public void setProgramme(String programme) {

this.programme = programme;

}

}

public class TestPerson {

public static void main(String[] args) {

// Create a Student object

Student student = new Student("John Doe", 1001, "Computer Science");

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

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

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

// Create a Lecturer object

Lecturer lecturer = new Lecturer("Dr. Smith", 2001, "Mathematics");

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

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

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

}

}

4. 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);

}

}