

A.J.Pavithra

28658

## **Java Practical 4**

### 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.

```
// Employee class

public class Employee {

    // Private variables

    private int empID;

    private String empName;

    private String empDesignation;


    // Getters for empID, empName, and empDesignation

    public int getEmpID() {

        return empID;

    }


    public String getEmpName() {

        return empName;

    }


    public String getEmpDesignation() {

        return empDesignation;

    }

}
```

```
// Setters for empID, empName, and empDesignation

public void setEmpID(int empID) {

    this.empID = empID;

}

public void setEmpName(String empName) {

    this.empName = empName;

}

public void setEmpDesignation(String empDesignation) {

    this.empDesignation = empDesignation;

}

}

// EmployeeTest class to test the code

public class EmployeeTest {

    public static void main(String[] args) {

        // Create two objects for Mr.Bogdan and Ms.Bird

        Employee bogdan = new Employee();

        bogdan.setEmpID(101);

        bogdan.setEmpName("Mr. Bogdan");

        bogdan.setEmpDesignation("Manager");

        Employee bird = new Employee();

        bird.setEmpID(102);

        bird.setEmpName("Ms. Bird");

        bird.setEmpDesignation("Engineer");

    }

}
```

```
// Print employee details using getters

    System.out.println("Employee Details for Mr. Bogdan:");

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

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

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


    System.out.println("\nEmployee Details for Ms. Bird:");

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

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

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

}

}
```

**Output:**

Employee Details for Mr. Bogdan:

ID: 101

Name: Mr. Bogdan

Designation: Manager

Employee Details for Ms. Bird:

ID: 102

Name: Ms. Bird

Designation: Engineer

### 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:

21

14

Explanation:

1. We have two classes, **SuperB** and **SubC**, where **SubC** extends **SuperB**.
2. **SuperB** has an instance variable **x**, and four methods:
  - **setIt**: sets the value of **x**.
  - **increase**: increments the value of **x** by 1.
  - **triple**: multiplies the value of **x** by 3.
  - **returnIt**: returns the value of **x**.
3. **SubC** overrides the **triple** method from its superclass (**SuperB**) and also has a new method called **quadruple**.
4. In the **main** method of **TestInheritance** class, we create objects **b** and **c** of classes **SuperB** and **SubC**, respectively.
5. For object **b**:
  - **setIt(2)** sets the value of **x** to 2.
  - **increase()** increments the value of **x** by 1, so **x** becomes 3.
  - **triple()** multiplies the value of **x** by 3, so **x** becomes 9.
  - **returnIt()** returns the value of **x**, which is 9.
6. For object **c**:
  - **setIt(2)** sets the value of **x** to 2.
  - **increase()** increments the value of **x** by 1, so **x** becomes 3.
  - **triple()** overrides the **triple** method from the superclass, so **x** becomes 6 (3 + 3).
  - **returnIt()** returns the value of **x**, which is 6.

Hence, the output is 21 (from object **b**) and 14 (from object **c**).

### 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	Lecturer	Person
- name	- name	Identify field and attributes to be stored in this class
- id	- id	
- course	- programme	
+ setName()/getName()	+ setName()/getName()	
+ setID()/getID()	+ setID()/getID()	
+ setCourse()/getCourse()	+ setProg()/getProg()	

```
// Person class
class Person {
    // Common attributes for both Student and Lecturer
    private String name;
    private int id;

    // Constructors for Person
    public Person() {
    }

    public Person(String name, int id) {
        this.name = name;
        this.id = id;
    }
}
```

```
// Getters and setters for name and 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 of Person)
```

```
class Student extends Person {  
    // Specific attribute for Student  
    private String course;
```



```
// Constructor for Student

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

    // Call the superclass constructor using 'super'

    super(name, id);

    this.course = course;

}


// Getter and setter for course

public String getCourse() {

    return course;

}


public void setCourse(String course) {

    this.course = course;

}

}
```

```
// Lecturer class (subclass of Person)

class Lecturer extends Person {

    // Specific attribute for Lecturer

    private String programme;


    // Constructor for Lecturer

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

        // Call the superclass constructor using 'super'

        super(name, id);

        this.programme = programme;

    }

    // Getter and setter for programme

    public String getProgramme() {

        return programme;

    }

    public void setProgramme(String programme) {

        this.programme = programme;

    }

}

// Test class to invoke Student and Lecturer

public class TestPerson {

    public static void main(String[] args) {

        // Create a Student object

        Student student = new Student("John Doe", 101, "Computer Science");
```

```
// Create a Lecturer object

    Lecturer lecturer = new Lecturer("Jane Smith", 201, "Software Engineering");


// Test the methods of 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("\nLecturer Details:");

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

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

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

}

}
```

Output:

```
Student Details:

Name: John Doe

ID: 101

Course: Computer Science

Lecturer Details:

Name: Jane Smith

ID: 201

Programme: Software 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);
```

```
    }
```

```
}
```

```
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
```