**Experiment Number : 05** **Date: 05.02.2026**

---

## INHERITANCE:

**Aim:**

To understand and implement inheritance concepts in Java.

**PRE LAB EXERCISE:**

**QUESTIONS:**

- ✓ **What is inheritance?**

  Inheritance is an object-oriented programming concept in which one class (child or subclass) acquires the properties and behaviors (fields and methods) of another class (parent or superclass). It helps in creating a hierarchical relationship between classes.

- ✓ **What is code reusability?**

  Code reusability means using existing code again instead of writing the same code repeatedly. It reduces duplication, saves time, and makes programs easier to maintain. Inheritance is one of the main features that supports code reusability in object-oriented programming.

- ✓ **What is the use of extends keyword?**

  The extends keyword is used to implement inheritance in Java. It allows a subclass to inherit the properties and methods of a superclass.

  **Example:**

```
class Parent {
   void show() {
      System.out.println("Hello");
   }
}

class Child extends Parent {
}
```

  Here, the Child class inherits the show() method from the Parent class.

**IN LAB EXERCISE:**

**Objective:**

To implement all types of inheritance.

**PROGRAMS:**

**Student Result System (Single Inheritance)**

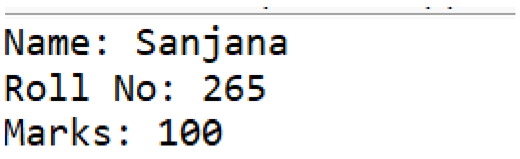
**Question:**
A school wants to store student details and calculate marks. Create a base class Student and a derived class Result.

**Code:**

```
class Student {
   String name;
   int rollNo;

   void getDetails() {
      name = "Sanjanaa";
      rollNo = 265;
   }
}

class Result extends Student {
   int marks = 100;

   void display() {
      System.out.println("Name: " + name);
      System.out.println("Roll No: " + rollNo);
      System.out.println("Marks: " + marks);
   }
}
```

```
public class Main {
    public static void main(String[] args) {
        Result r = new Result();
        r.getDetails();
        r.display();
    }
}
```

**Output:**



```
Name: Sanjana
Roll No: 265
Marks: 100
```

2. **Bank Account System (Hierarchical Inheritance)**

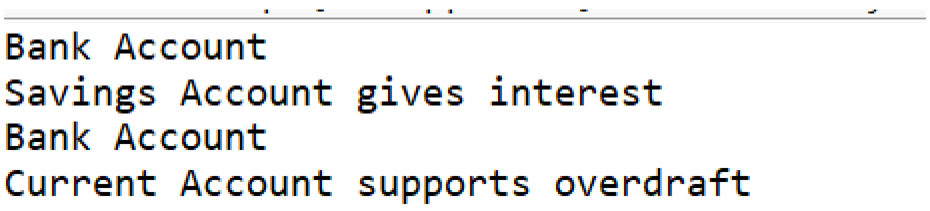
**Question:**
A bank has Savings and Current accounts. Both inherit from a common Account class.

**Code:**

```
class Account {
    void showAccountType() {
        System.out.println("Bank Account");
    }
}


class SavingsAccount extends Account {
    void interest() {
        System.out.println("Savings Account gives interest");
    }
}
```

```
class CurrentAccount extends Account {
    void overdraft() {
        System.out.println("Current Account supports overdraft");
    }
}

public class Main {
    public static void main(String[] args) {
        SavingsAccount s = new SavingsAccount();
        CurrentAccount c = new CurrentAccount();

        s.showAccountType();
        s.interest();

        c.showAccountType();
        c.overdraft();
    }
}
```

**Output:**



```
Bank Account
Savings Account gives interest
Bank Account
Current Account supports overdraft
```

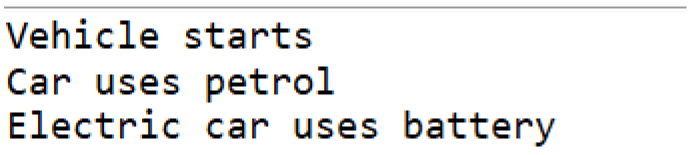
**3. Vehicle System (Multilevel Inheritance)**

**Question:**
A company classifies vehicles as Vehicle → Car → ElectricCar.

**Code:**

```
class Vehicle {
    void start() {
        System.out.println("Vehicle starts");
    }
}

class Car extends Vehicle {
    void fuelType() {
        System.out.println("Car uses petrol");
    }
}

class ElectricCar extends Car {
    void battery() {
        System.out.println("Electric car uses battery");
    }
}

public class Main {
    public static void main(String[] args) {
        ElectricCar e = new ElectricCar();
        e.start();
        e.fuelType();
        e.battery();
    }
}
```

**Output:**



```
Vehicle starts
Car uses petrol
Electric car uses battery
```

**POST LAB EXERCISE:**

- ✓ **Why Java does not support multiple inheritance using classes and how it is implemented?**

  Java does not support multiple inheritance using classes to avoid ambiguity and complexity, such as the Diamond Problem. Instead, Java achieves multiple inheritance using interfaces.

- ✓ **What is the role of the super keyword? Give examples.**

  The super keyword is used to refer to the parent class. It is used to access parent class variables, methods, and constructors.

- ✓ **Can a child class access private members of the parent class? Why?**

  No, a child class cannot access private members of the parent class because private members are restricted to the class in which they are declared. They can be accessed indirectly through public or protected methods.

- ✓ **Explain why hybrid inheritance is not supported in Java.**

  Hybrid inheritance is not supported using classes in Java because it can cause ambiguity and complexity. However, it is supported using interfaces.

**Result:**

Thus the different types of inheritance were implemented and executed successfully.

**ASSESSMENT**

| **Description** | **Max Marks** | **Marks Awarded** |
|---|---|---|
| Pre Lab Exercise | **5** | |
| In Lab Exercise | **10** | |
| Post Lab Exercise | **5** | |
| Viva | **10** | |
| **Total** | **30** | |
| **Faculty Signature** | | |