Experiment Number : 05 **Date: 05-02-2026**

INHERITANCE

Aim:

To understand and implement inheritance concepts in Java.

PRE LAB EXERCISE

QUESTIONS

- ✓ What is inheritance?

  Answer: Inheritance is an object-oriented concept where a **child class acquires the properties and methods of a parent class** using the extends keyword. It helps reuse existing code.

- ✓ What is code reusability?

  Answer: Code reusability means **using the same code multiple times without rewriting it**. In Java, inheritance allows code reusability by letting child classes use parent class methods.

- ✓ What is the use of extends keyword?

  Answer: The extends keyword is used to **inherit one class from another class**. It allows a subclass to access the variables and methods of the superclass.

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:

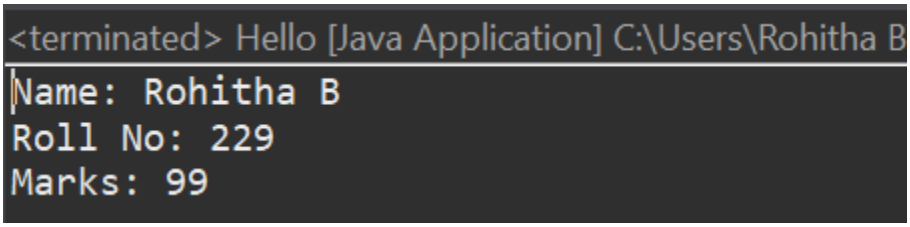
```
class Student {
    String name;
    int rollNo;

    void getDetails() {
        name = "Rohitha B";
        rollNo = 229;
    }
}

class Result extends Student {
    int marks = 99;

    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: Rohitha B

Roll No: 229

Marks: 99



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();
    }
}
```

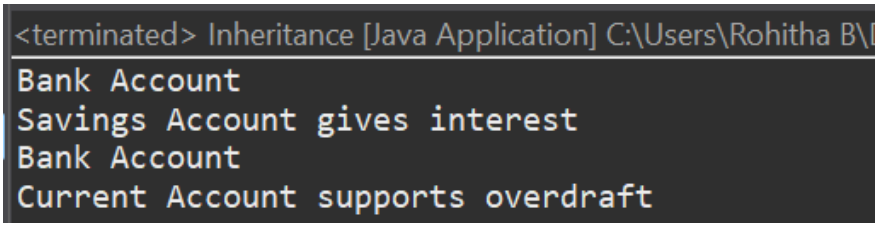
Output:

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

```
<terminated> Inheritance [Java Application] C:\Users\Rohitha B\
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?

  Answer: Java does not support multiple inheritance using classes to avoid ambiguity problems like the **Diamond Problem**, where the compiler gets confused if two parent classes have the same method.

  Instead, Java implements multiple inheritance using **interfaces**, because interfaces contain only method declarations, avoiding confusion.

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

  Answer: The super keyword is used to:

1. Access parent class variables
2. Call parent class methods
3. Call parent class constructor

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

  Answer: No, a child class **cannot access private members** of the parent class because private members are accessible **only within the same class**. This supports data hiding and security.

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

  Answer: Hybrid inheritance combines multiple inheritance and other types, which can again lead to ambiguity and complexity. Since Java does not support multiple inheritance using classes, **hybrid inheritance is also not supported directly**. However, it can be achieved 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	

ROHITHA B

24BCS229