Experiment Number : 05 **Date:**

INHERITANCE

Aim:

To understand and implement inheritance concepts in Java.

PRE LAB EXERCISE

QUESTIONS

1.What is inheritance?

- Inheritance is an object-oriented programming concept where a child class acquires the properties and methods of a parent class.
- It helps in code reuse and establishes a relationship between classes.
- Inheritance improves program structure and maintainability.

2.What is code reusability?

- Code reusability means using existing code in multiple places instead of writing it again.
- It reduces code duplication and saves development time.
- Inheritance, functions, and libraries help achieve code reusability.
- What is the use of extends keyword?

3.The extends keyword is used to create a child class from a parent class.

- It allows the child class to inherit variables and methods of the parent class.
- It is commonly used in anguages like Java and C++.

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 = "Anitha";
        rollNo = 101;
    }
}

class Result extends Student {
    int marks = 85;

    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: RAM

Roll No: 101

Marks: 85

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();
    }
}
```

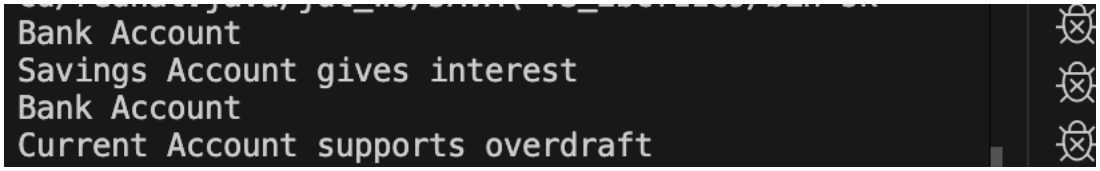
Output:

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

1.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.
- If two parent classes have the same method, the compiler cannot decide which one to inherit.
- Java implements multiple inheritance using interfaces, where method conflicts are avoided.

2.What is the role of the super keyword? Give examples.

- The super keyword is used to refer to the parent class object.
- It is used to access parent class variables, methods, and constructors.
-
- Example:
- super.display(); → calls parent class method
- super(); → calls parent class constructor

3.Can a child class access private members of the parent class? Why?

- No, a child class cannot directly access private members of the parent class.

- Private members are restricted to the same class only.
- They can be accessed indirectly using public or protected methods (getters/setters).

4.Explain why hybrid inheritance is not supported in Java.

- Hybrid inheritance involves a combination of multiple inheritance types, which may cause ambiguity.
- Since Java does not support multiple inheritance with classes, hybrid inheritance is also not supported.
- However, Java can achieve hybrid inheritance 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		