INHERITANCE

Aim:

To understand and implement inheritance concepts in Java.

PRE LAB EXERCISE

QUESTIONS

- ✓ **What is inheritance?**

  Inheritance is an OOP concept where one class (child/subclass) acquires the properties and methods of another class (parent/superclass). It lets you build new classes based on existing ones.

- ✓ **What is code reusability?**

  Code reusability means using existing code again instead of rewriting it. It reduces duplication, saves time, and makes programs easier to maintain.

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

  extends keyword is used to create inheritance. It tells the compiler that one class is derived from another and should inherit its features.

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();
```

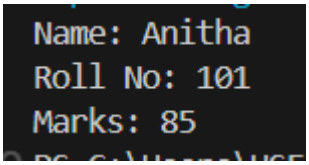
```
        r.getDetails();
        r.display();
    }
}
```

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
```

```
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");
    }
}
```

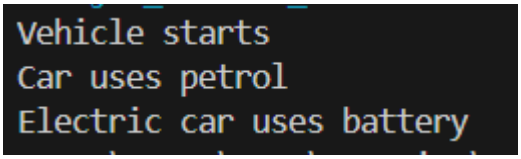
```
public class Main {
    public static void main(String[] args) {
        ElectricCar e = new ElectricCar();
        e.start();
        e.fuelType();
        e.battery();
    }
}
```

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?**

  Because it creates confusion if two parent classes have the same method (diamond problem). Java avoids this error. Instead, it uses interfaces to achieve multiple inheritance safely.

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

super is used to refer to the parent class.

Uses :

- Call parent constructor → super();
- Access parent variable → super.x
- Call parent method → super.show();

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

No. Private members belong only to the parent class. This protects data (encapsulation).

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

Hybrid inheritance includes multiple inheritance. Since Java doesn't allow multiple inheritance with classes, hybrid inheritance with classes is also not supported. Interfaces are used instead.

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	