Lab 5: To illustrate the concept of Polymorphism

Objectives:

To understand the concept of polymorphism.

Polymorphism:

Polymorphism is the ability of an object to take on different forms. In Java, polymorphism refers to the ability of a class to provide different implementations of a method, depending on the type of object that is passed to the method.

Method overriding:

- Method overriding allows a subclass to provide its own implementation for a method that is already defined in the superclass.
- When a method in a subclass has the same name, parameters or signature, as a method in its super-class, then the method in the subclass (the child class) overrides the method in the super-class (the parent class).
- It enables polymorphism, where a single method name can be used to invoke different implementations based on the object's actual type

Rules for Java Method Overriding

1. The method must have the same name as in the parent class.
2. The method must have the same parameter as in the parent class.
3. There must be an IS-A relationship (inheritance).

Super Keyword in method overriding:

- The super keyword is used for calling the parent class method/constructor. super.myMethod() calls the myMethod() method of base class
- super() calls the constructor of base class.

Program 1:

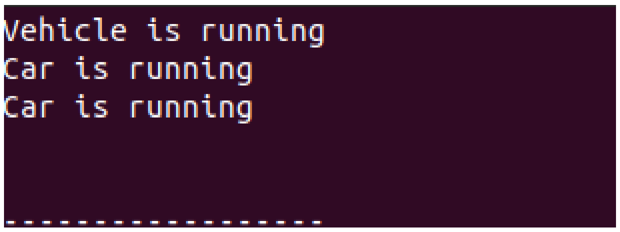
```
class Vehicle
 {
   void run()
        {
        System.out.println("Vehicle is running");
        }
        }

class Car extends Vehicle
        {
        void run()
        {
        System.out.println("Car is running");
        }
        public static void main(String args[])
        {

         Vehicle v =new Vehicle();
        v.run();
        //creating an instance of child class
        Vehicle v1 =new Car();
        v1.run();
         Car obj = new Car();
        //calling the method with child class instance
        obj.run();


        }
        }
```

Output:



```
Vehicle is running
Car is running
Car is running


-----------------
```

2. Create a class Animal with the method makeSound() that represents a sound. Also, extend class Dog from class Animal with the same method. Also, extend class Cat from the class Dog with the same method. Using the concept of method overriding and super keyword and necessary variables, access the methods from main class Mammal.

3. Create a superclass Shape with a method to display(). Derive a class Rectangle from it and make a method to display(). Again derive a class Triangle from class Shape. Use the concept of method overriding and access the members from the main class Test.

4.Create a base class Student with attributes as name and rollno with a method displayDetails(). Derive a subclass StudentExam from the base class with additional attribute marks. Override the displayDetails() method in the StudentExam to display both the student details along with their scores.

5.Create a class Employee with necessary data members and methods. Take the input from the user to create instances of different types of employees. Derive the class PermanentEmployee and ContractEmployee from the superclass Employee with appropriate members and methods from the super class. Display the details of the employees and display salary for each employee. Use the concept of method overriding as appropriate.