Polymorphism

1.Create a base class called Vehicle with the following methods: ● void start(): This method should print "Vehicle started." Create two subclasses of Vehicle called Car and Motorcycle. Override the start() method in each subclass to provide a specific implementation: ● Car: Print "Car started." ● Motorcycle: Print "Motorcycle started." Create a class called Garage with a method named serviceVehicle(Vehicle vehicle). Inside this method, call the start() method of the provided vehicle object and print "Vehicle serviced." In the Main class, create instances of Car and Motorcycle. Create an instance of the Garage class. Call the serviceVehicle() method of the Garage class with instances of both Car and Motorcycle.

```
public class Vehicle {

	void start() {
		System.out.println("Vehicle started");
	}

}

class Car extends Vehicle{

	@Override
	void start() {
		System.out.println("Car started");
	}
}


class Motorcycle extends Vehicle{

	@Override
	void start() {
		System.out.println("Motorcycle started");
	}
```

```
}

class Garage {

        void serviceVehicle(Vehicle vehicle) {
                vehicle.start();
      System.out.println("Vehicle serviced.");
        }
}

public class VehicleMain {

        public static void main(String[] args) {
      Car car = new Car();
      Motorcycle motorcycle = new Motorcycle();
      Garage garage = new Garage();

      System.out.println("Service for Car:");
      garage.serviceVehicle(car);

      System.out.println("\nService for Motorcycle:");
      garage.serviceVehicle(motorcycle);
  }

}
```

2.Create a class called Student. Inside the Student class, implement the following instance variables (fields): ● String name ● int age ● String department Implement the following constructors in the Student class: ● A default constructor that initializes the name to "Unknown", age to 20, and department to "Unassigned". ● A constructor that takes two parameters: name and age, and initializes the department to "IT". ● A constructor that takes three parameters: name, age, and department. In the Main class, create instances of the Student class using each constructor. Printoutthedetailsofeachstudent, including their name, age, and department

```
public class Student {

	String name;
	int age;
	String department;

	public Student() {
		this.name = "unkonown";
		this.age= 20;
		this.department= "Unassigned";
	}

	public Student(String name, int age) {
    // Constructor with name and age parameters
    this.name = name;
    this.age = age;
    this.department = "IT";
  }

	public Student(String name, int age, String department) {
    // Constructor with name, age, and department parameters
    this.name = name;
    this.age = age;
    this.department = department;
  }

	// Method to print student details
  public void printDetails() {
    System.out.println("Name: " + name);
    System.out.println("Age: " + age);
    System.out.println("Department: " + department);
  }

}

public class StudentMain {
	public static void main(String[] args) {
    // Create instances of the Student class using different constructors
    Student student1 = new Student(); // Default constructor
    Student student2 = new Student("John", 25); // Constructor with name and age
    Student student3 = new Student("Alice", 22, "Engineering"); // Constructor with name, age,
and department

    // Print details of each student
```

```
        System.out.println("Details of Student 1:");
        student1.printDetails();

        System.out.println("\nDetails of Student 2:");
        student2.printDetails();

        System.out.println("\nDetails of Student 3:");
        student3.printDetails();
    }
}
```

By - Divya Parihar