ASSIGNMENT 1 :

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

SOURCE CODE :

```
// Base class Vehicle
class Vehicle {
    void start() {
        System.out.println("Vehicle started.");
    }
}

// Subclass Car
class Car extends Vehicle {
    @Override
    void start() {
        System.out.println("Car started.");
    }
}

// Subclass Motorcycle
class Motorcycle extends Vehicle {
    @Override
    void start() {
        System.out.println("Motorcycle started.");
    }
}

// Garage class
class Garage {
    void serviceVehicle(Vehicle vehicle) {
        vehicle.start();  // Polymorphic call
        System.out.println("Vehicle serviced.");
    }
}
```

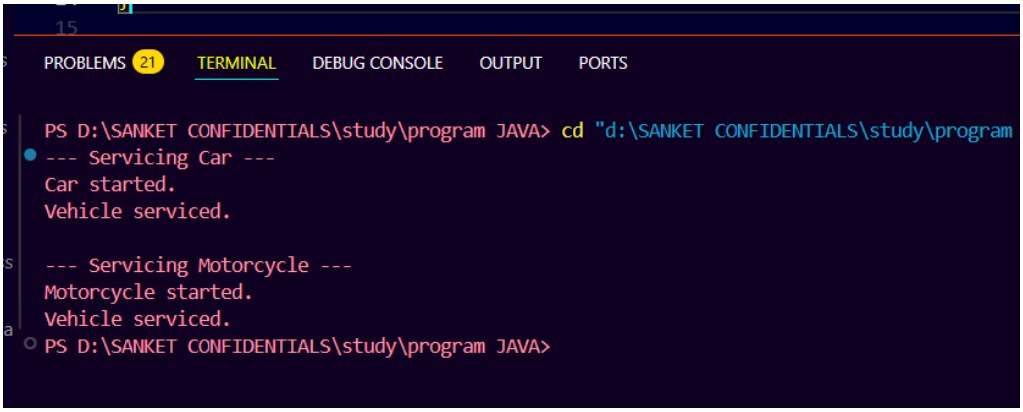
```java
// Main class
public class Main {
    public static void main(String[] args) {
        Car myCar = new Car();
        Motorcycle myMotorcycle = new Motorcycle();

        Garage garage = new Garage();

        System.out.println("--- Servicing Car ---");
        garage.serviceVehicle(myCar);

        System.out.println("\n--- Servicing Motorcycle ---");
        garage.serviceVehicle(myMotorcycle);
    }
}
```

OUTPUT:



ASSIGNMENT 2:

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: ● Adefault constructor that initializes the name to "Unknown", age to 20, and department to "Unassigned". ● Aconstructor that takes two parameters: name and age, and initializes the department to "IT". ● Aconstructor that takes three parameters: name, age, and department. In the Main class, create instances of the Student class using each constructor. Print out the details of each student, including their name, age, and department

SOURCE CODE:

```
class Student {
   String name;
   int age;
   String department;

   // default constructor
   Student() {
      name = "Unknown";
      age = 20;
      department = "Unassigned";
   }

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

   // constructor with all three
   Student(String name, int age, String department) {
      this.name = name;
      this.age = age;
      this.department = department;
   }

   void printDetails() {
      System.out.println("Name: " + name);
      System.out.println("Age: " + age);
      System.out.println("Department: " + department);
      System.out.println();
   }
}
```

```
public class lab5_que2 {
   public static void main(String[] args) {

      Student s1 = new Student(); // default
      Student s2 = new Student("Sanket", 21); // 2 args
      Student s3 = new Student("Riya", 22, "Computer"); // all 3 args

      System.out.println("Student 1:");
      s1.printDetails();

      System.out.println("Student 2:");
      s2.printDetails();

      System.out.println("Student 3:");
      s3.printDetails();
   }
}
```

OUTPUT :