

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.

Program:

```
//Base class
```

```
class Vehicle{
```

```
    //Method to strt the vehicle
```

```
    public void start(){
```

```
        System.out.println("Vehicle Started."); // Base class implementation
```

```
    }
```

```
}
```

```
// Subclass Car
```

```
class Car extends Vehicle{
```

```
    //Override start method for car
```

```
    @Override
```

```
    public void start(){
```

```
        System.out.println("Car started."); // Override the start method for car
```

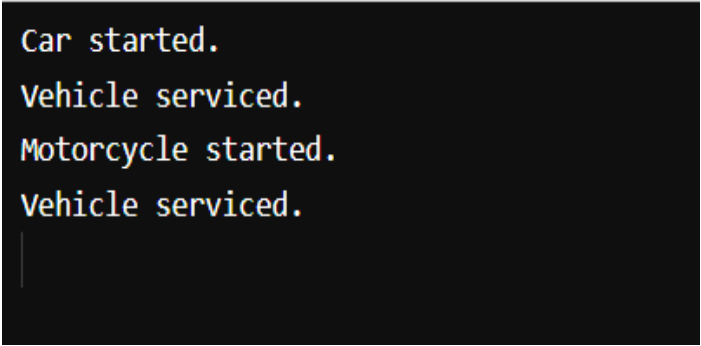
```

    }
}
//Subclass Motorcycle
class Motorcycle extends Vehicle {
    //Override start method for Motorcycle
    @Override
    public void start() {
        System.out.println("Motorcycle started."); // Override the start method for
        Motorcycle
    }
}
class Garage {
    //Method to service a vehicle
    public void serviceVehicle(Vehicle vehicle) {
        vehicle.start(); // Call the start method of the provided vehicle
        System.out.println("Vehicle serviced.");
    }
}
//Main class
public class Main {
    public static void main(String[] args) {
        //Create instance of car and Motorcycle
        Car car = new Car();
        Motorcycle motorcycle = new Motorcycle();
        Garage garage = new Garage(); // Service the car
    }
}

```

```
garage.serviceVehicle(car);  
garage.serviceVehicle(motorcycle); // Service the motorcycle  
}  
}
```

Output:

A dark-themed terminal window showing the output of the code. The text is as follows:

```
Car started.  
Vehicle serviced.  
Motorcycle started.  
Vehicle serviced.  
|
```

The vertical bar at the end of the last line indicates the cursor position.

Car started.
Vehicle serviced.
Motorcycle started.
Vehicle serviced.

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

Program:

```
class Student {  
  
    // Instance variables (fields) String name; int age; String department;  
  
    // Default constructor  
    public Student() {  
        name = "Unknown";  
        age = 20;  
        department = "Unassigned";  
    }  
  
    // Constructor with name and age parameters  
    public Student(String name, int age) {  
        this.name = name;  
        this.age = age;  
        this.department = "IT"; // Set department to "IT"  
    }  
  
    // Constructor with name, age, and department parameters
```

```

public Student(String name, int age, String department) {
    this.name = name;
    this.age = age;
    this.department = department;
}

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

}

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

    // Create instances of Student using different constructors

    Student student1 = new Student(); // Using default constructor

    Student student2 = new Student("Cock", 26);

    // Using constructor with name and age

    Student student3 = new Student("Sid", 25, "Information Science");

    // Using constructor with name, age, and department

    // Display student information
    student1.displayInfo();
    System.out.println();
    student2.displayInfo();
    System.out.println();
    student3.displayInfo();
}

}

```

Output:

Name: Unknown

Age: 20

Department: Unassigned

Name: Cock

Age: 26

Department: IT

Name: Sid

Age: 25

Department: Information Science