

**Name: Khushi Satyaprakash Singh**

**Branch: Information Technology**

**Div: B**

**Roll number: 45**

**Batch: [B-2]**

---

### **Assignment Number :10**

```
abstract class Vehicle { // Renamed Car to Vehicle
    public Vehicle(VehicleType type) { // Renamed model to type
        this.type = type;
        arrangeParts();
    }

    private void arrangeParts() {
    }

    protected abstract void construct();

    private VehicleType type = null;

    public VehicleType getType() {
        return type;
    }

    public void setType(VehicleType type) {
        this.type = type;
    }

    @Override
    public String toString() { // Added toString method for meaningful output
        return "Vehicle type: " + type;
    }
}

class VehicleFactory { // Renamed CarFactory to VehicleFactory
    public static Vehicle buildVehicle(VehicleType type) { // Renamed Car to Vehicle and
model to type
        Vehicle vehicle = null;
        switch (type) {
            case SMALL:
                vehicle = new SmallVehicle();
        }
    }
}
```

```

        break;
    case SEDAN:
        vehicle = new SedanVehicle();
        break;
    case LUXURY:
        vehicle = new LuxuryVehicle();
        break;
    default:
        break;
    }
    return vehicle;
}
}

```

```

enum VehicleType { // Renamed CarType to VehicleType
    SMALL, SEDAN, LUXURY
}

```

```

class LuxuryVehicle extends Vehicle { // Renamed LuxuryCar to LuxuryVehicle
    LuxuryVehicle() {
        super(VehicleType.LUXURY);
        construct();
    }

    @Override
    protected void construct() {
        System.out.println("Building a luxury vehicle");
    }
}

```

```

class SedanVehicle extends Vehicle { // Renamed SedanCar to SedanVehicle
    SedanVehicle() {
        super(VehicleType.SEDAN);
        construct();
    }

    @Override
    protected void construct() {
        System.out.println("Building a sedan vehicle");
    }
}

```

```

class SmallVehicle extends Vehicle { // Renamed SmallCar to SmallVehicle
    SmallVehicle() {

```

```

        super(VehicleType.SMALL);
        construct();
    }

    @Override
    protected void construct() {
        System.out.println("Building a small vehicle");
    }
}

class TestFactoryPattern {
    public static void main(String[] args) {
        System.out.println(VehicleFactory.buildVehicle(VehicleType.SMALL));

        System.out.println(VehicleFactory.buildVehicle(VehicleType.SEDAN));

        System.out.println(VehicleFactory.buildVehicle(VehicleType.LUXURY));
    }
}

```

**Output:**

```

Building a small vehicle
Vehicle type: SMALL
Building a sedan vehicle
Vehicle type: SEDAN
Building a luxury vehicle
Vehicle type: LUXURY

```