OOPL Assignment – 9

Factory Design Pattern

Name :- Atharva Kinikar

Div : SE10

Batch : F10

Roll No :23241

Code :-

1] Car.java

/\*

Name:-Atharva Kinikar

Div :- SE10

Batch :- F10

Roll No :23241

\*/

public abstract class Car {

    public Car(CarType model, ManufacturerType manufacturer) {

        this.model = model;

        this.manufacturer = manufacturer;

        arrangeParts();

    }

    private void arrangeParts() {

        System.out.println("Car Manufacturing Started");

    }

    protected abstract void construct();

    private CarType model = null;

    private ManufacturerType manufacturer = null;

    public CarType getModel() {

        return model;

    }

    public void setModel(CarType model) {

        this.model = model;

    }

    public ManufacturerType getManufacturer() {

        return manufacturer;

    }

    public void setManufacturer(ManufacturerType manufacturer) {

        this.manufacturer = manufacturer;

    }

}

2] CarFactory.java

/\*

Name:-Atharva Kinikar

Div :- SE10

Batch :- F10

Roll No :23241

\*/

public class CarFactory {

    public static Car buildCar(CarType model, ManufacturerType manufacturer) {

        Car car = null;

        switch (model) {

            case SMALL:

                car = new SmallCar();

                break;

            case SEDAN:

                car = new SedanCar();

                break;

            case LUXURY:

                car = new LuxuryCar();

                break;

            default:

                System.out.println("Invalid Choice !!!");

                break;

        }

        return car;

    }

}

3] CarType.java

public enum CarType{

    SMALL, SEDAN, LUXURY

}

4] LuxuryCar.java

public class LuxuryCar extends Car{

    LuxuryCar(){

        super(CarType.LUXURY, ManufacturerType.BMW);

        construct();

    }

    @Override

    protected void construct() {

        System.out.println("Building the Luxury car");

        System.out.println("Model Name : BMW 520d");

    }

}

5] SedanCar.java

public class SedanCar extends Car{

    SedanCar(){

        super(CarType.SEDAN, ManufacturerType.SKODA);

        construct();

    }

    @Override

    protected void construct() {

        System.out.println("Building the Sedan");

        System.out.println("Model Name : Skoda Octavia");

    }

}

6] SmallCar.java

public class SmallCar extends Car{

    SmallCar(){

        super(CarType.SMALL, ManufacturerType.HYUNDAI);

        construct();

    }

    @Override

    protected void construct() {

        System.out.println("Building Small Car ");

        System.out.println("Model Name : Hyundai i20");

    }

}

7] TestFactoryPattern.java

public class TestFactoryPattern{

    public static void main(String[] args) {

        System.out.println("CAR FACTORY\n");

        System.out.println(CarFactory.buildCar(CarType.SMALL, ManufacturerType.HYUNDAI));

        System.out.println();

        System.out.println(CarFactory.buildCar(CarType.SEDAN, ManufacturerType.SKODA));

        System.out.println();

        System.out.println(CarFactory.buildCar(CarType.LUXURY, ManufacturerType.BMW));

    }

}