**ASSIGNMENT-10**

**package** Car;

**enum** CarType {

***small***,***sedan***,***luxury***

}

**public** **abstract** **class** Car {

**public** Car(CarType model) {

**this**.model = model;

arrangeParts();

}

**private** **void** arrangeParts() {

System.***out***.println("building a car");

}

**protected** **abstract** **void** construct();

**private** CarType model = **null**;

**public** CarType getModel() {

**return** model;

}

**public** **void** setModel(CarType model) {

**this**.model = model;

}

}

**class** LuxuryCar **extends** Car {

LuxuryCar() {

**super**(CarType.***luxury***);

construct();

}

**protected** **void** construct() {

System.***out***.println("Building luxury car");

// add accessories

}

}

**class** SedanCar **extends** Car {

SedanCar() {

**super**(CarType.***sedan***);

construct();

}

**protected** **void** construct() {

System.***out***.println("Building sedan car");

}

}

**class** SmallCar **extends** Car {

SmallCar() {

**super**(CarType.***small***);

construct();

}

**protected** **void** construct() {

System.***out***.println("Building small car");

}

}

**class** CarFactory {

**public** **static** Car buildCar(CarType model) {

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 input");

**break**;

}

**return** car;

}

}

**class** TestFactoryPattern {

**public** **static** **void** main(String[] args)

{

System.***out***.println(CarFactory.*buildCar*(CarType.***small***));

System.***out***.println(CarFactory.*buildCar*(CarType.***sedan***));

System.***out***.println(CarFactory.*buildCar*(CarType.***luxury***));

}

}