Assignment - 10 - (OOP)

**public** **interface** Car{

**void** createChassis();

**void** assembleParts();

**void** paintCar();

**void** checkWorking();

**void** safetyTest();

}

**public** **class** HatchBack **implements** Car{

@Override

**public** **void** createChassis() {

System.out.println("Creating Chassis for Hatchback");

}

@Override

**public** **void** assembleParts() {

System.out.println("Assembling parts for Hatchback");

}

@Override

**public** **void** paintCar() {

System.out.println("Painting the Hatchback");

}

@Override

**public** **void** checkWorking() {

System.out.println("Hatchback is Running : No defects");

}

@Override

**public** **void** safetyTest() {

System.out.println("Hatchback Crash test : Car is safe");

}

}

**public** **class** Sedan **implements** Car{

@Override

**public** **void** createChassis() {

System.out.println("Creating Chassis for Sedan");

}

@Override

**public** **void** assembleParts() {

System.out.println("assembling parts for Sedan");

}

@Override

**public** **void** paintCar() {

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

}

@Override

**public** **void** checkWorking() {

System.out.println("Sedan is Running Properly: No defects");

}

@Override

**public** **void** safetyTest() {

System.out.println("Sedan Crash test : Car is safe");

}

}

**public** **class** SUV **implements** Car{

@Override

**public** **void** createChassis() {

System.out.println("Creating Chassis for SUV");

}

@Override

**public** **void** assembleParts() {

System.out.println("assembling parts for SUV");

}

@Override

**public** **void** paintCar() {

System.out.println("painting the SUV");

}

@Override

**public** **void** checkWorking() {

System.out.println("SUV is Running Properly: No defects");

}

@Override

**public** **void** safetyTest() {

System.out.println("SUV Crash test : Car is safe");

}

}

**public** **class** CarFactory {

**public** Car getCar(String TypeOfCar){

**if**(TypeOfCar == **null**){

**return** **null**;

}

**if**(TypeOfCar.equalsIgnoreCase("Hatchback")){

**return** **new** Hatchback();

} **else** **if**(TypeOfCar.equalsIgnoreCase("Sedan")){

**return** **new** Sedan();

} **else** **if**(TypeOfCar.equalsIgnoreCase("SUV")){

**return** **new** SUV();

}

**return** **null**;

}

}

**public** **class** TestFactoryPattern {

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

CarFactory obj = **new** CarFactory();

Car a = obj.getCar("Hatchback");

a.gatherParts();

a.assembleParts();

a.paintCar();

a.checkRunning();

a.safetyTest();

Car b = obj.getCar("Sedan");

b.gatherParts();

b.assembleParts();

b.paintCar();

b.checkRunning();

b.safetyTest();

Car c = obj.getCar("SUV");

c.gatherParts();

c.assembleParts();

c.paintCar();

c.checkRunning();

c.safetyTest();

}

}