```
package Factory;
import java.util.Scanner;
import java.awt.Color;
public interface Car // Car interface defining the blueprint for car objects
{
  public String getModel();
  public void setWheel(String wheel);
  public String getWheel();
  public void setEngine(String engine);
  public String getEngine();
  public String getColour();
  public void setColour(String colour);
  public String getVariant();
  public void setVariant(String variant);
  public String getFuel();
  public void setFuel(String fuel);
  public String getdata();
}
public interface CarFactory // CarFactory interface defining the factory method for building cars
{
    public abstract Car buildCar(String model, String wheel, String engine,String colour,String
variant,String fuel);
}
public class HatchbackCar implements Car // Implementation of a Hatchback car
{
     String model,wheel,engine,fuel,variant,colour;
     HatchbackCar(String model, String wheel, String engine,String colour,String variant,String fuel)
  {
         this.model = model;
   this.wheel = wheel;
   this.engine = engine;
   this.fuel = fuel;
   this.variant = variant;
   this.colour = colour;
     }
  public String getModel()
  {
   return model;
  }
  public void setWheel(String wheel)
  {
   this.wheel = wheel;
  }
  public String getWheel()
  {
   return wheel;
  }
  public void setEngine(String engine)
  {
   this.engine = engine;
  }
  public String getEngine()
  {
   return engine;
  }
  public String getColour()
  {
   return colour;
  }
  public void setColour(String colour)
  {
   this.colour = colour;
  }
  public String getVariant()
  {
   return variant;
  }
  public void setVariant(String variant)
  {
   this.variant = variant;

  }
  public String getFuel()
  {
   return fuel;
  }
  public void setFuel(String fuel)
  {
   this.fuel = fuel;

  }
  public String getdata()
  {
   if(model.equals("NA"))
   {
    return "Car not built";
   }
   String data = "Model = "+model+"\nVariant = "+variant+"\nEngine = "+engine+"\nFuel type =
"+fuel+"\nColour = "+colour+"\nTyres Compound = "+wheel;
   return data;
  }
}
public class HatchbackCarFactory implements CarFactory // Factory class for creating HatchbackCar
objects
{
  public Car buildCar(String model, String wheel, String engine,String colour,String variant,String fuel)
  {
        Car car = new HatchbackCar(model, wheel, engine,colour,variant,fuel);
        return car;
  }
}
public class SedanCar implements Car // Implementation of a Sedan car
{
 String model,wheel,engine,fuel,variant,colour;
    SedanCar(String model, String wheel, String engine,String colour,String variant,String fuel)
 {
     this.model = model;
  this.wheel = wheel;
  this.engine = engine;
  this.fuel = fuel;
  this.variant = variant;
  this.colour = colour;
    }
 public String getModel()
 {
  return model;
 }
 public void setWheel(String wheel)
 {
  this.wheel = wheel;
 }
 public String getWheel()
 {
  return wheel;
 }
 public void setEngine(String engine)
 {
  this.engine = engine;
 }
 public String getEngine()
 {
  return engine;
 }
 public String getColour()
 {
  return colour;
 }
 public void setColour(String colour)
 {
  this.colour = colour;
 }
 public String getVariant()
 {
  return variant;
 }
 public void setVariant(String variant)
 {
  this.variant = variant;
 }
 public String getFuel()
 {
  return fuel;
 }
 public void setFuel(String fuel)
 {
  this.fuel = fuel;
 }
 public String getdata()
 {
  if(model.equals("NA"))
  {
   return "Car not built";
  }
  String data = "Model = "+model+"\nVariant = "+variant+"\nEngine = "+engine+"\nFuel type =
"+fuel+"\nColour = "+colour+"\nTyres Compound = "+wheel;
```

```
  return data;
 }
}
public class SedanCarFactory implements CarFactory // Factory class for creating SedanCar objects
{
 public Car buildCar(String model, String wheel, String engine,String colour, String variant,String fuel)
    {
        Car car = new SedanCar(model, wheel, engine,colour,variant,fuel);
        return car;
 }
}
public class SUVCar implements Car // Implementation of an SUV car
{
   String model,wheel,engine,fuel,variant,colour;
   SUVCar(String model, String wheel, String engine,String colour,String variant,String fuel)
   {
      this.model = model;
   this.wheel = wheel;
   this.engine = engine;
   this.fuel = fuel;
   this.variant = variant;
   this.colour = colour;
   }
  public String getModel()
  {
   return model;
  }
  public void setWheel(String wheel)
  {
   this.wheel = wheel;
  }
  public String getWheel()
  {
   return wheel;
  }
  public void setEngine(String engine)
  {
   this.engine = engine;
  }
  public String getEngine()
  {
   return engine;
  }
  public String getColour()
  {
   return colour;
  }
  public void setColour(String colour)
  {
   this.colour = colour;
  }
  public String getVariant()
  {
   return variant;
  }
  public void setVariant(String variant)
  {
   this.variant = variant;
  }
  public String getFuel()
  {
   return fuel;
  }
  public void setFuel(String fuel)
  {
   this.fuel = fuel;
  }
  public String getdata()
  {
   if(model.equals("NA"))
   {
    return "Car not built";
   }
   String data = "Model = "+model+"\nVariant = "+variant+"\nEngine = "+engine+"\nFuel type = 
"+fuel+"\nColour = "+colour+"\nTyres Compound = "+wheel;
   return data;
  }
}
public class SUVCarFactory implements CarFactory // Factory class for creating SUVCar objects.
{
 public Car buildCar(String model, String wheel, String engine,String colour,String variant,String fuel)
 {
       Car car = new SUVCar(model, wheel, engine,colour,variant,fuel);
       return car;
 }
}
public class TestFactoryPattern // Main class to test the factory pattern for car creation.
{
 CarFactory carBuilder;
 Car car;
    public static void main(String[] args)
    {
     TestFactoryPattern client = new TestFactoryPattern();
        client.buildCarMethod();
    }
    public void buildCarMethod()
    {
     int ui;
     Scanner sc = new Scanner (System.in);
     System.out.println("Enter your choice:");
     System.out.println("1.)Hatchback  2.)Sedan  3.)SUV");
     ui = sc.nextInt();
     switch(ui)
     {
       case 1: carBuilder = new HatchbackCarFactory();
         System.out.println("Choice of hatckback is  1.)Audi RS3  2.)Audi RS6  3.)Audi A1");
         ui=sc.nextInt();
         switch(ui)
         {
          case 1: car = carBuilder.buildCar("Audi RS3", "Medium", "2.5L Turbocharged Inline
5","Blue","Sport","Petrol");
            break;
          case 2: car = carBuilder.buildCar("Audi RS6", "Soft", "6.0L Twin Turbo V8","Dark
Gray","GT","E85");
            break;
          case 3: car = carBuilder.buildCar("Audi A1", "Hard", "2.0L Turbocharged Inline 4","Abyss
Black","N8","Diesel");
            break;
          default:System.out.println("Invalid option");
            car = carBuilder.buildCar("NA", "NA", "NA","NA","NA","NA");
            break;
         }
         break;
       case 2: carBuilder = new SedanCarFactory();
         System.out.println("Choice of Sedan is  1.)BMW M5 Competition  2.)BMW M3 Competition  3.)BMW
Alpina B7");
         ui=sc.nextInt();
         switch(ui)
         {
          case 1: car = carBuilder.buildCar("BMW M5 Competition", "Very Soft", "4.4-liter Twin-Turbocharged
V8","Pearl White","VDi","Petrol");
            break;
          case 2: car = carBuilder.buildCar("BMW M3 Competition", "Soft", "3.0-liter Twin-Turbocharged
Inline-6","Obsidian Blue","ZX","Petrol");
            break;
          case 3: car = carBuilder.buildCar("BMW Alpina B7", "Medium", "4.4-liter Twin-Turbocharged
V8","Metallic Silver","VX CVT","Petrol");
            break;
          default:System.out.println("Invalid option");
            car = carBuilder.buildCar("NA", "NA", "NA","NA","NA","NA");
            break;
         }
         break;
       case 3: carBuilder = new SUVCarFactory();
         System.out.println("Choice of SUV is  1.)Porsche Cayenne Turbo GT  2.)Porsche Macan GTS
3.)Porsche Cayenne Turbo S E-Hybrid");
         ui=sc.nextInt();
         switch(ui)
         {
          case 1: car = carBuilder.buildCar("Porsche Cayenne Turbo GT", "Soft", "4.0-liter
Twin-Turbocharged V8","Magma Grey","VXI CNG","Petrol");
            break;
          case 2: car = carBuilder.buildCar("Porsche Macan GTS", "Soft", "2.9-liter Twin-Turbocharged
V6","Lunar Silver Metallic","ZX","Petrol");
            break;
          case 3: car = carBuilder.buildCar("Porsche Cayenne Turbo S E-Hybrid", "Soft", "4.0-liter
Twin-Turbocharged V8 with Electric Motor","Platinum White Pearl","Legender 4x4","Hybrid");
            break;
          default:System.out.println("Invalid option");
            car = carBuilder.buildCar("NA", "NA", "NA","NA","NA","NA");
            break;
         }
         break;
     }
        System.out.println(car.getdata());
```

```
    }
}
```

Output:

```
Enter your choice:
1.)Hatchback  2.)Sedan  3.)SUV
3
Choice of SUV is  1.)Porsche Cayenne Turbo GT  2.)Porsche Macan GTS  3.)Porsche Cayenne Turbo S
E-Hybrid
1
Model = Porsche Cayenne Turbo GT
Variant = VXI CNG
Engine = 4.0-liter Twin-Turbocharged V8
Fuel type = PetrolColour = Magma Grey
Tyres Compound = Soft
```