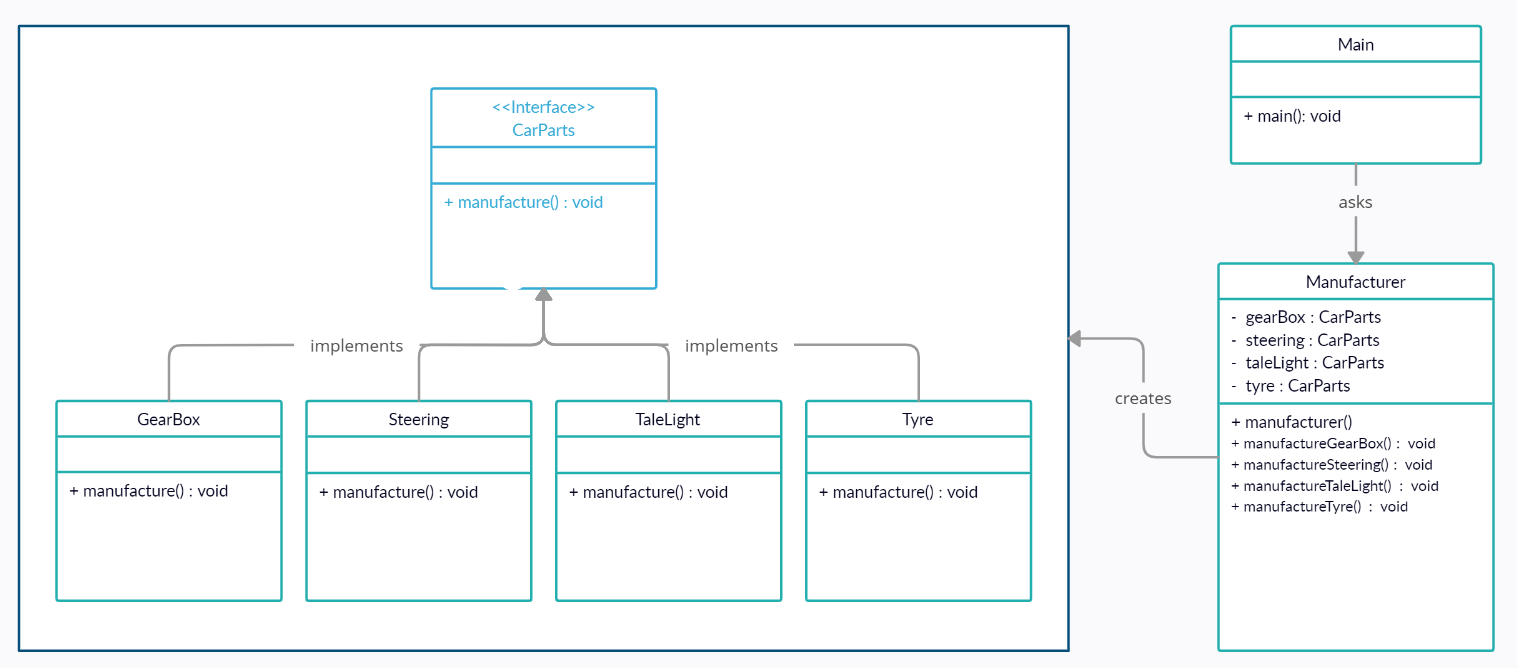
**Assignment 8: Facade Design Pattern**

**What is Facade Design Pattern?**

**Facade** is a structural design pattern that provides a simplified interface to a library, a framework, or any other complex set of classes.

**Structure (Class Diagram)**

****

**Implementation (Code)**

public class *Steering* implements *CarParts* {  
 public void *manufacture*() {  
 *System*.out.*println*("Steering Manufacturing in Progress\nManufacturing Process Completed ✅");  
 }  
}

public interface *CarParts* {  
 void *manufacture*();  
}

public class *GearBox* implements *CarParts* {  
 public void *manufacture*() {  
 *System*.out.*println*("Gear Box Manufacturing in Progress\nManufacturing Process Completed ✅");  
 }  
}

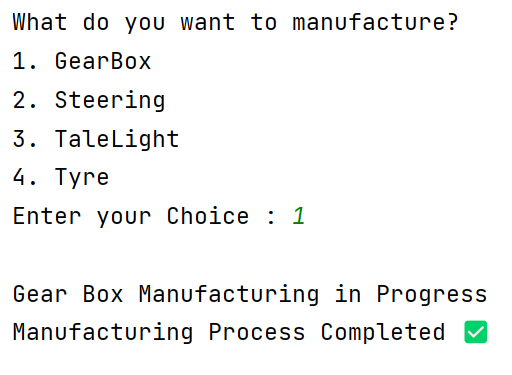
public class *TaleLight* implements *CarParts* {  
 public void *manufacture*() {  
 *System*.out.*println*("Tale Light Manufacturing in Progress\nManufacturing Process Completed ✅");  
 }  
}

public class *Tyre* implements *CarParts* {  
 public void *manufacture*() {  
 *System*.out.*println*("Tyre Manufacturing in Progress\nManufacturing Process Completed ✅");  
 }  
}

public class *Manufacturer* {  
 private *CarParts* gearBox;  
 private *CarParts* steering;  
 private *CarParts* taleLight;  
 private *CarParts* tyre;  
  
 public *Manufacturer*() {  
 gearBox = new *GearBox*();  
 steering = new *Steering*();  
 taleLight = new *TaleLight*();  
 tyre = new *Tyre*();  
 }  
  
 public void *manufactureGearBox*() {  
 gearBox.*manufacture*();  
 }  
 public void *manufactureStreering*() {  
 steering.*manufacture*();  
 }  
 public void *manufactureTaleLight*() {  
 taleLight.*manufacture*();  
 }  
 public void *manufactureTyre*() {  
 tyre.*manufacture*();  
 }  
}

**Output**

import *java.util.Scanner*;  
  
public class *Main* {  
 public static void *main*(*String*[] *args*) {  
 *Scanner* sc = new *Scanner*(*System*.in);  
 *Manufacturer* manufacturer = new *Manufacturer*();  
  
 *System*.out.*println*("What do you want to manufacture? ");  
 *System*.out.*println*("1. GearBox \n2. Steering \n3. TaleLight \n4. Tyre");  
 *System*.out.*print*("Enter your Choice : ");  
 int choice = sc.*nextInt*();  
 *System*.out.*println*();  
  
 switch (choice) {  
 case (1) -> manufacturer.*manufactureGearBox*();  
 case (2) -> manufacturer.*manufactureStreering*();  
 case (3) -> manufacturer.*manufactureTaleLight*();  
 case (4) -> manufacturer.*manufactureTyre*();  
 default -> *System*.out.*println*("Enter a Valid Choice!");  
 }  
  
 }  
}



**Applicability**

1. Use the Facade pattern when you need to have a limited but straightforward interface to a complex subsystem.
2. Use the Facade when you want to structure a subsystem into layers.