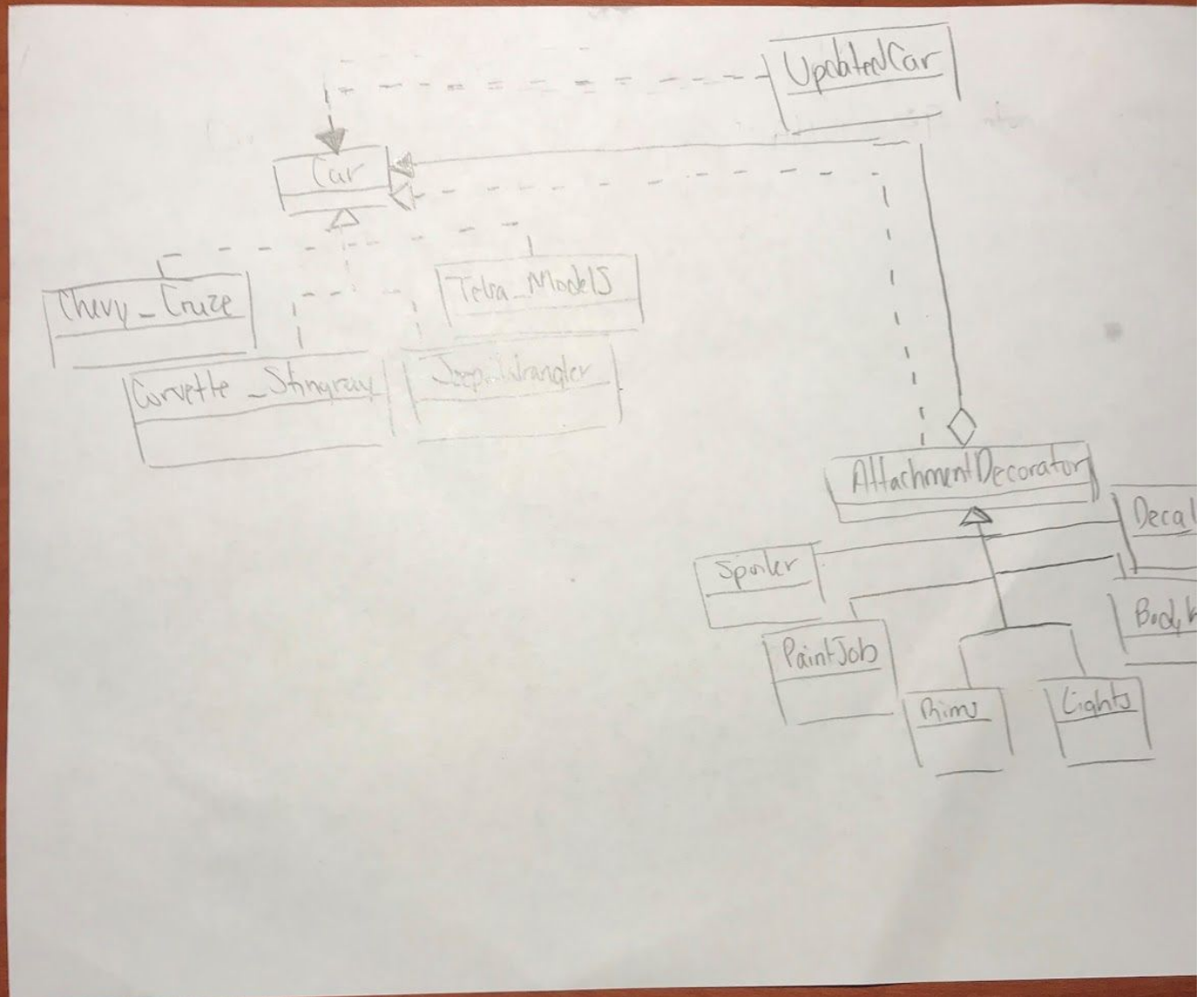


Name: Dominique Cooper  
Eagle ID: 900929361  
Course: CSCI 5335

## UML Diagram:



---

## **Car Interface and 4 implementations:**

```
public interface Car {  
    public void carType();  
}
```

---

```
public class Chevy_Cruze implements Car{  
    public void carType() {  
        System.out.println("Car being modified: Chevy Cruze");  
    }  
}
```

---

```
public class Corvette_Stingray implements Car {  
    public void carType() {  
        System.out.println("Car being modified: Corvette Stingray");  
    }  
}
```

---

```
public class Jeep_Wrangler implements Car {  
    public void carType() {  
        System.out.println("Car being modified: Jeep Wrangler");  
    }  
}
```

---

```
public class Tesla_ModelS implements Car{  
    public void carType() {  
        System.out.println("Car being modified: Tesla Model S");  
    }  
}
```

---

## **Abstract Class and also implementation of Car class, AttachmentDecorator and its six extended classes:**

```
public abstract class AttachmentDecorator implements Car {
```

protected Car attachment;

```
public AttachmentDecorator(Car attachment) {  
    this.attachment = attachment;
```

```
}
```

```
public void carType() {  
    attachment.carType();
```

```
}
```

```
}
```

---

```
public class BodyKit extends AttachmentDecorator {
```

```
    public BodyKit(Car attachment) {  
        super(attachment);
```

```
    }
```

```
    public void carType() {  
        attachment.carType();  
        setBodyKit(attachment);
```

```
    }
```

```
    private void setBodyKit(Car attachment) {  
        System.out.println("Modified attachment obtained: BodyKit");
```

```
    }
```

```
}
```

---

```
public class Decal extends AttachmentDecorator {
```

```
    public Decal(Car attachment) {  
        super(attachment);
```

```
    }
```

```
    public void carType() {  
        attachment.carType();  
        setDecal(attachment);
```

```
    }
```

```
    private void setDecal(Car attachment) {  
        System.out.println("Modified attachment obtained: Decal");
```

```
    }
```

```
}
```

---

```
public class Decal extends AttachmentDecorator {
    public Decal(Car attachment) {
        super(attachment);
    }
    public void carType() {
        attachment.carType();
        setDecal(attachment);
    }

    private void setDecal(Car attachment) {
        System.out.println("Modified attachment obtained: Decal");
    }
}
```

---

```
public class Lights extends AttachmentDecorator {
    public Lights(Car attachment) {
        super(attachment);
    }
    public void carType() {
        attachment.carType();
        setLights(attachment);
    }

    private void setLights(Car attachment) {
        System.out.println("Modified attachment obtained: Lights");
    }
}
```

---

```
public class PaintJob extends AttachmentDecorator {
    public PaintJob(Car attachment) {
        super(attachment);
    }
    public void carType() {
        attachment.carType();
        setPaintJob(attachment);
    }

    private void setPaintJob(Car attachment) {
        System.out.println("Modified attachment obtained: PaintJob");
    }
}
```

---

```

public class Rims extends AttachmentDecorator{
    public Rims(Car attachment) {
        super(attachment);
    }
    public void carType() {
        attachment.carType();
        setRims(attachment);
    }

    private void setRims(Car attachment) {
        System.out.println("Modified attachment obtained: Rims");
    }
}

```

---

```

public class Spoiler extends AttachmentDecorator {
    public Spoiler(Car attachment) {
        super(attachment);
    }
    public void carType() {
        attachment.carType();
        setSpoiler(attachment);
    }

    private void setSpoiler(Car attachment) {
        System.out.println("Modified attachment obtained: Spoiler");
    }
}

```

---

## Test and Compile UpdatedClass:

```

public class UpdatedCar {
    public static void main(String[] args) {
        Car ChevyCruze = new BodyKit(new Decal(new PaintJob(new Chevy_Cruze())));
        ChevyCruze.carType();
        System.out.println("=====");

        Car CorvetteStingray = new BodyKit(new Rims(new PaintJob(new Lights(new Spoiler(new
        Corvette_Stingray()))));
        CorvetteStingray.carType();
        System.out.println("=====");
    }
}

```

```

Car JeepWrangler = new Rims(new PaintJob(new Lights(new Jeep_Wrangler())));
JeepWrangler.carType();
System.out.println("=====");

Car TeslaModelS = new Rims(new Lights(new Tesla_ModelS()));
TeslaModelS.carType();
System.out.println("=====");
}
}

```

---

## Working Java Code:

