LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CS4001NI Programming

Assessment Weightage & Type
30% Individual Coursework

Year and Semester

2021 - 22 Spring - 2

Student Name: Shishir Ghimire

London Met ID:

College ID: NP01CP4S220018

Assignment Due Date: 2022-05-20

Assignment Submission Date: 2022-05-20

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION:

The first coursework of this module is to develop a system by creating three classes for the vehicle to display detail, book, purchase, and sell the vehicle. My IDE for this coursework was BlueJ. My coursework consists of three classes and they are Vehicle, AutoRickshaw, and ElectricScooter. Here Vehicle class was my parent class and AutoRickshaw and ElectricScooter were sub-classes of Vehicle class. Different methods were created during this coursework and accessor methods were assigned for each class and the various display methods were created to display the data of every class.

While working on this coursework I had to use different tools and software to complete the given course. Some of those are listed below:

BlueJ: BlueJ is the IDE that allows users to develop Java programs quickly and easily. Simple, Interactive, Portable, Mature, and innovative are its main features. I used BlueJ to develop my java program consisting of my three classes Vehicle, AutoRickshaw, and ElectricScooter. (Unknown, n.d.)

Figure 21: BlueJ Logo

MS Word: Released on October 25, 1983, Microsoft Word is the word processing software developed by Microsoft cooperation. I used MS Word to write my report consisting of a class diagram, pseudocode, method description, testing, error, and

its correction. It is a very useful and handy to write a reports of any coursework. (Wikipedia , 2022)

Figure 22: MS Word Logo

Draw.io: It is a free and open-source platform. It is mainly used for graphs and drawing. I used this tool to create my class diagram which was very handy using draw.io. (Wekipedia, 2022)

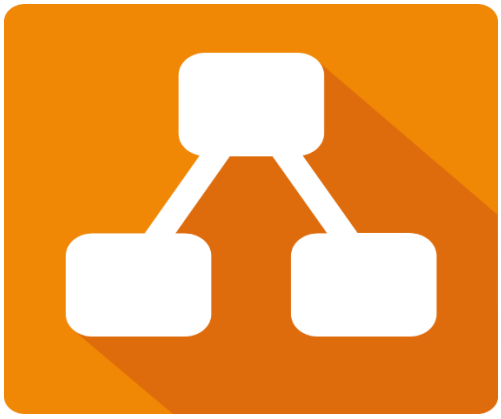

Figure 23: draw.io Logo

2. CLASS DIAGRAM:

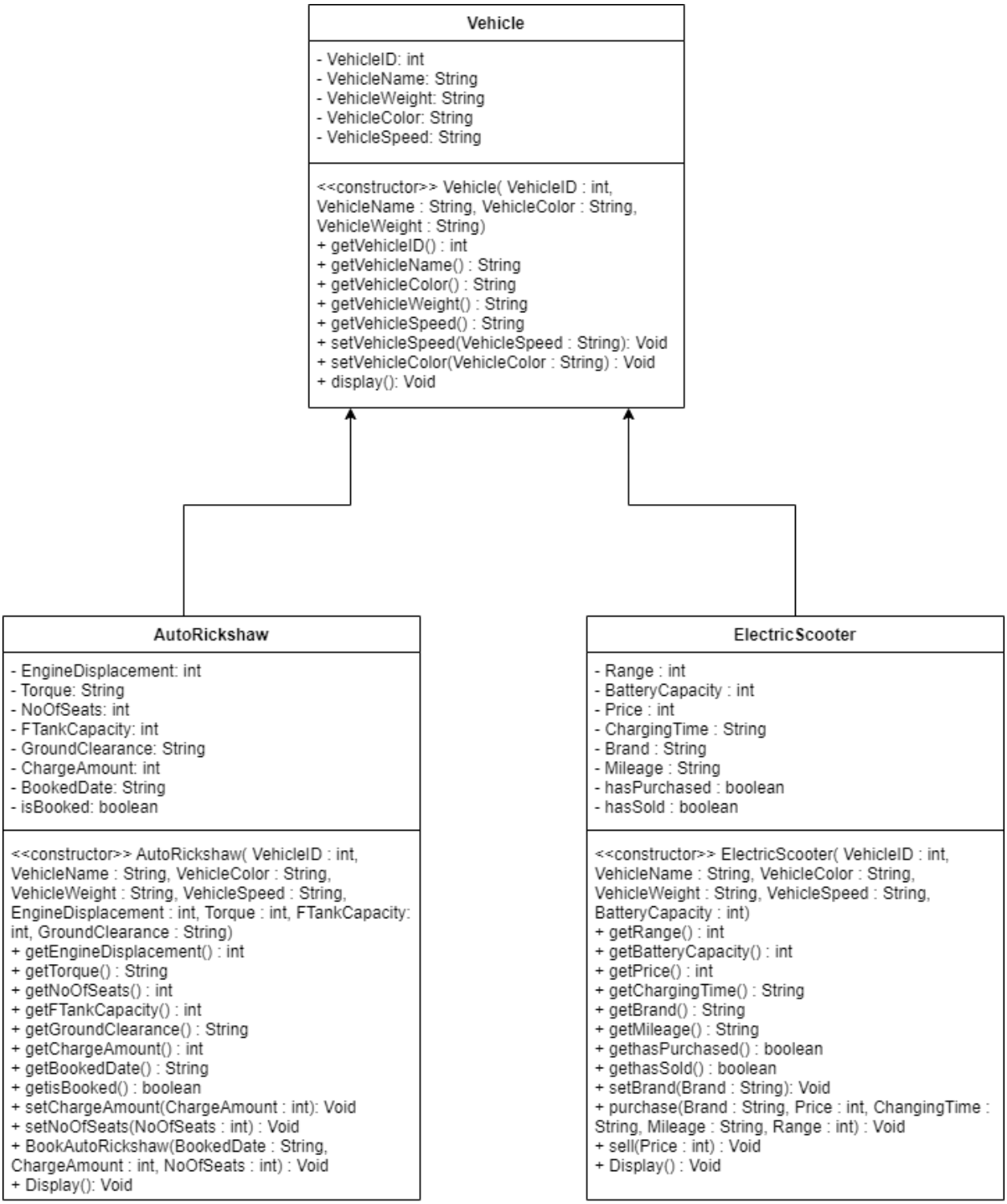

Figure 24: Class Diagram

3. Pseudocode:

3.1. Pseudocode for Vehicle:

```
CREATE METHOD getVehicleID()
        DO
                Return VehicleID
        END DO
CREATE METHOD getVehicleName()
        DO
                Return VehicleName
        END DO
CREATE METHOD getVehicleColor()
        DO
                Return VehicleColor
        END DO
CREATE METHOD getVehicleWeight()
        DO
                Return VehicleWeight
        END DO
CREATE METHOD getVehicleSpeed()
        DO
                Return VehicleSpeed
        END DO
CREATE METHOD SetVehicleSpeed(String VehicleSpeed)
        DO
                this.VehicleSpeed=VehicleSpeed
        END DO
CREATE METHOD setVehicleColor(String VehicleColor)
        DO
                this.VehicleColor=VehicleColor
        END DO
```

```
CREATE METHOD display()
    DO
        DISPLAY “The Vehicle ID is”, VehicleID
        DISPLAY “The Vehicle Name is”, VehicleName
        DISPLAY “The Vehicle Color is”, VehicleColor
        DISPLAY “The Vehicle Weight is”, VehicleWeight
        DISPLAY “The Vehicle Speed is”, VehicleSpeed
    IF (VehicleWeight == “”) THEN
        DISPLAY “This is empty”
    ELSE
        DISPLAY “The Vehicle Weight is”, VehicleWeight
    END IF
    END DO
```

3.2. Pseudocode for AutoRickshaw:

```
CREATE METHOD getEngineDisplacement()
    DO
        Return EngineDisplacement
    END DO
CREATE METHOD getTorque()
    DO
        Return Torque
    END DO
CREATE METHOD getNoOfSeats()
    DO
        Return NoOfSeats
    END DO
CREATE METHOD getFTankCapacity()
    DO
        Return FTankCapacity
    END DO
```

```
CREATE METHOD getGroundClearance()
        DO
                Return GroundClearance
        END DO
CREATE METHOD getChargeAmount()
        DO
                Return BookedDate
        END DO
CREATE METHOD getisBooked()
        DO
                Return IsBooked
        END DO
CREATE METHOD setChargeAmount(INT ChargeAmount)
        DO
                this.ChargeAmount=ChargeAmount
        END DO
CREATE METHOD setVehicleColor(int NoOfSeats)
        DO
                this.NoOfSeats=NoOfSeats
        END DO
CREATE METHOD BookAutoRickshaw (String BookedDate, int ChargedAmount,
int NoOfSeats)
        DO
                IF (ISBooked == true) THEN
                        DISPLAY “The Auto Rickshaw is booked.”
                        DISPLAY “Status: “, IsBooked
                ELSE
                        this.IsBooked = true
                        this.setChargeAmount(ChargeAmount)
                        this.setNoOfSeats(NoOfSeats)
                        this.BookedDate = BookedDate
```

```
            END IF
        END DO
    CREATE METHOD Display()
        DO
        Super.display()
            IF (IsBooked == true)
            DISPLAY “The Engine Displacement is: “, EngineDisplacement
            DISPLAY “The torque is: “, Torque
            DISPLAY “The number of seats are: “ NoOfSeats
            DISPLAY “The total fuel tank capacity is:”, FTankCapacity
            DISPLAY “The Ground Clearance is:”, GroundClearance
            DISPLAY “The total charged amount is:”, ChargeAmount
            DISPLAY “The booking date is :”, Bookeddate
            END IF
        END DO
```

3.3. Pseudocode for ElectricScooter

```
CREATE METHOD getRange()
    DO
        Return Range
    END DO
CREATE METHOD getBatteryCapacity
    DO
        Return BatteryCapacity
    END DO
CREATE METHOD getPrice()
    DO
        Return Price
    END DO
CREATE METHOD getChargingTime()
    DO
```

```
            Return ChargingTime
        END DO

CREATE METHOD getBrand()
        DO
            Return Brand
        END DO
CREATE METHOD getMileage()
        DO
            Return Mileage
        END DO
CREATE METHOD getHasPurchased()
        DO
            Return hasPurchased
        END DO
CREATE METHOD getHasSold()
        DO
            Return hasSold
        END DO
CREATE METHOD setBrand(String Brand)
        DO
            IF (haspurchased != true)
            This.Brand = Brand
            ELSE
            DISPLAY “Brand cannot be changed as it’s already sold”
            END IF
        END DO
CREATE METHOD purchase (String Brand, int Price, String ChargingTime, String
Mileage, int Range)
        DO
            IF (hasPurchases == false)
```

```
                setBrand(Brand)
                this.hasPurchased = true
                this.Range = Range
                this.Price = Price
                this.ChargingTime = ChargingTime
                this.Mileage = Mileage
        ELSE
                DISPLAY “The Electric Scooter is already purchased.”
        END IF
    END DO
CREATE METHOD sell (int Price)
    DO
        IF (hasSold == false)
                this.Price = Price
                this.ChargingTime = " "
                this.Mileage = " "
                this.BatteryCapacity =0
                this.Range =0
                this.hasSold = true
                this.hasPurchased = false
        ELSE
                DISPLAY “The Vehicle is Sold out”
        END IF
    END DO
CREATE METHOD DISPLAY()
    DO
        Super.display();
        IF (hasPurchased==true)
        THEN
        DISPLAY “The electric scooter brand is: “, Brand
        DISPLAY “The battery capacity of electric scooter is: “ ,BatteryCapacity
```

DISPLAY “The electric scooter gives the mileage of: “ , Mileage
DISPLAY “The range of electric scooter is: “ , Range
DISPLAY “The charging time of scooter is: “ ,ChargingTime
END IF
END DO

4. METHOD DESCRIPTION:

The methods of all three classes are described below:

4.1. For Vehicle class:

getVehicleID()

This method is used here to get the value of VehicleID.

getVehicleName()

This method is used here to get the value of VehicleName.

getVehicleColor()

This method is used here to get the value of VehicleColor.

getVehicleWeight()

This method is used here to get the value of VehicleWeight.

getVehicleSpeed()

This method is used here to get the value of VehicleSpeed.

setVehicleSpeed(String VehicleSpeed)

This method is used to set the value of VehicleSpeed.

setVehicleColor(String VehicleColor)

This method is used to set the value of VehicleColor.

display()

This method is used to print out the VehicleID, VehicleName, VehicleColor, VehicleSpeed and if VehicleWeight is not empty, VehicleWeight too.

4.2. For AutoRickshaw class:

getEngineDisplacement()

This method is used to get the value of EngineDisplacement.

getTorque()

This method is used to get the value of Torque.

getNoOfSeats()

This method is used to get the value of NoOfSeats.

getFTankCapacity()

This method is used to get the value of FTankCapacity.

getGroundClearance()

This method is used to get the value of GroundClearance.

getChargeAmount()

This method is used to get the value of ChargeAmount.

getBookedDate()

This method is used to get the value of BookedDate.

getIsBooked()

This method is used to get the value of isBooked. It's value can be either true or false.

setChargeAmount(int ChargeAmount)

This method is used to set ChargeAmount value.

setNoOfSeats(int NoOfseats)

This method is used to set NoOfSeats value.

BookAutoRickshaw(String BookedDate, int ChargeAmount, int NoOfSeats)

This method is used to book an auto rickshaw. If the value of IsBooked is true, then this method displays The Auto Rickshaw is booked message with its status whether it is true or false. Else it sets the value ChargeAmount, NoOfSeats and BookedDate according to parameters and sets IsBooked to true.

Display()

This method displays display method from Vehicle class and it print out the value of EngineDisplacement, Torque, NoOfSeats, FTankCapacity, BookedDate, GroundClearance, ChargeAmount.

4.3. For ElectricScooter class:

getRange()

This method is used to get the value of Range.

getBatteryCapacity()

This method is used to get the value of BatteryCapacity.

getPrice()

This method is used to get the value of the Price.

getChargingTime()

This method is used to get the value of the ChargingTime.

getBrand()

This method is used to get the value of the Brand.

getMileage()

This method is used to get the value of the Mileage.

getHasPurchased

This method is used to get the value of hasPurchased in Boolean form.

getHasSold()

This method is used to get the value of hasSold in Boolean form.

setBrand(String Brand)

This method is used to set the value of Brand if the value of hasPurchased is not true it displays the brand else it prints Brand cannot be changed as it's already purchased.

Purchase (String Brand, int Price, String ChargingTime, String Mileage, int Range)

This method is used to check whether the Electric Scooter is already sold or not. If the value of hasPurchased is false then it sets the values of Brand, hasPurchased to true, Range, Price, ChargingTime, and Mileage.

sell (int Price)

This method is used to whether the scooter is sold out or not. If the value of hasSold is false then it sets the value of Price, ChargingTime to empty, Mileage to empty, BatteryCapacity is 0, Range to 0, hasSold to true, and hasPurchased to false. Else it prints out 'The Vehicle is Sold out.'

DISPLAY()

This method is used to display the display method of the Vehicle class. Also If the value of hasPurchased is set to be true it prints Brand, BatteryCapacity, Mileage, Range, and ChargingTime values.

5. Testing

Following are the testing for the given question:

5.1. Test 1

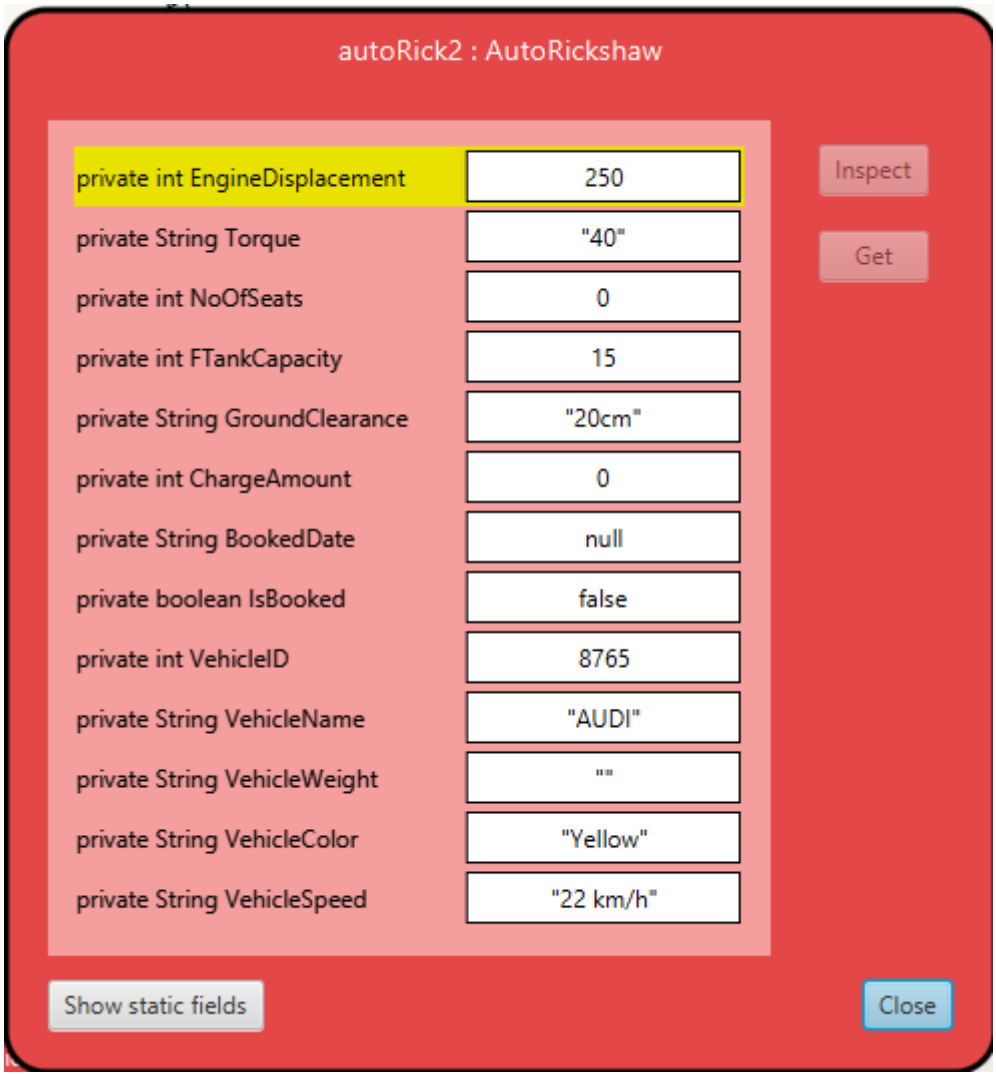

Figure 25: Inspecting AutoRickshaw Class

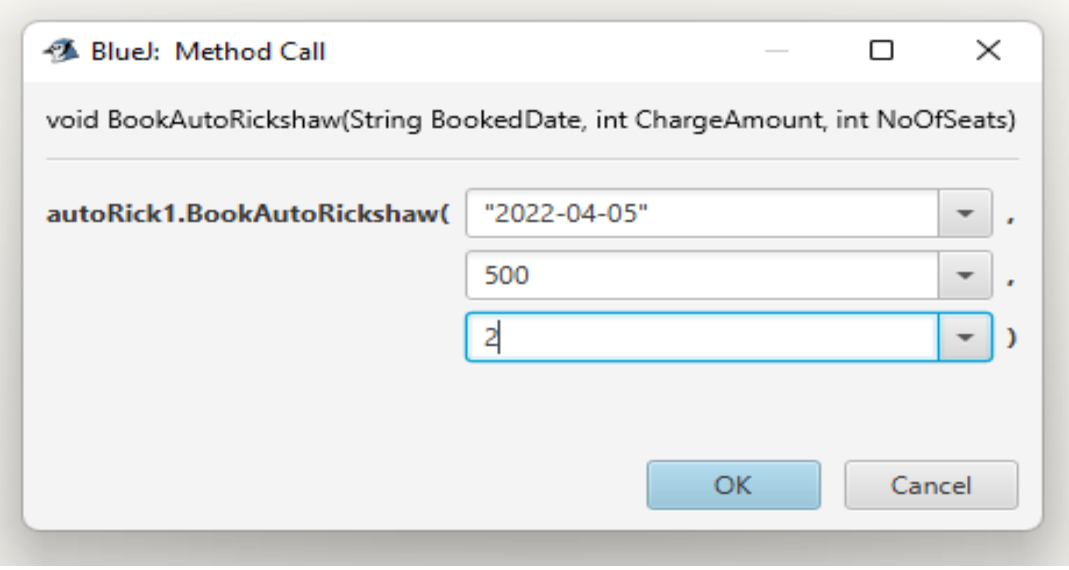

Figure 26: Booking Auto-Rickshaw

Figure 27: Re-Inspecting AutoRickshaw Class

Objective	To book an Auto-Rickshaw
Action	Inspect, book, re-inspect
Expected Result	The auto-Rickshaw will be booked.
Actual Result	The auto-Rickshaw has been appointed.
Conclusion	The test was successful.

Table 4: Test 1

5.2. Test 2

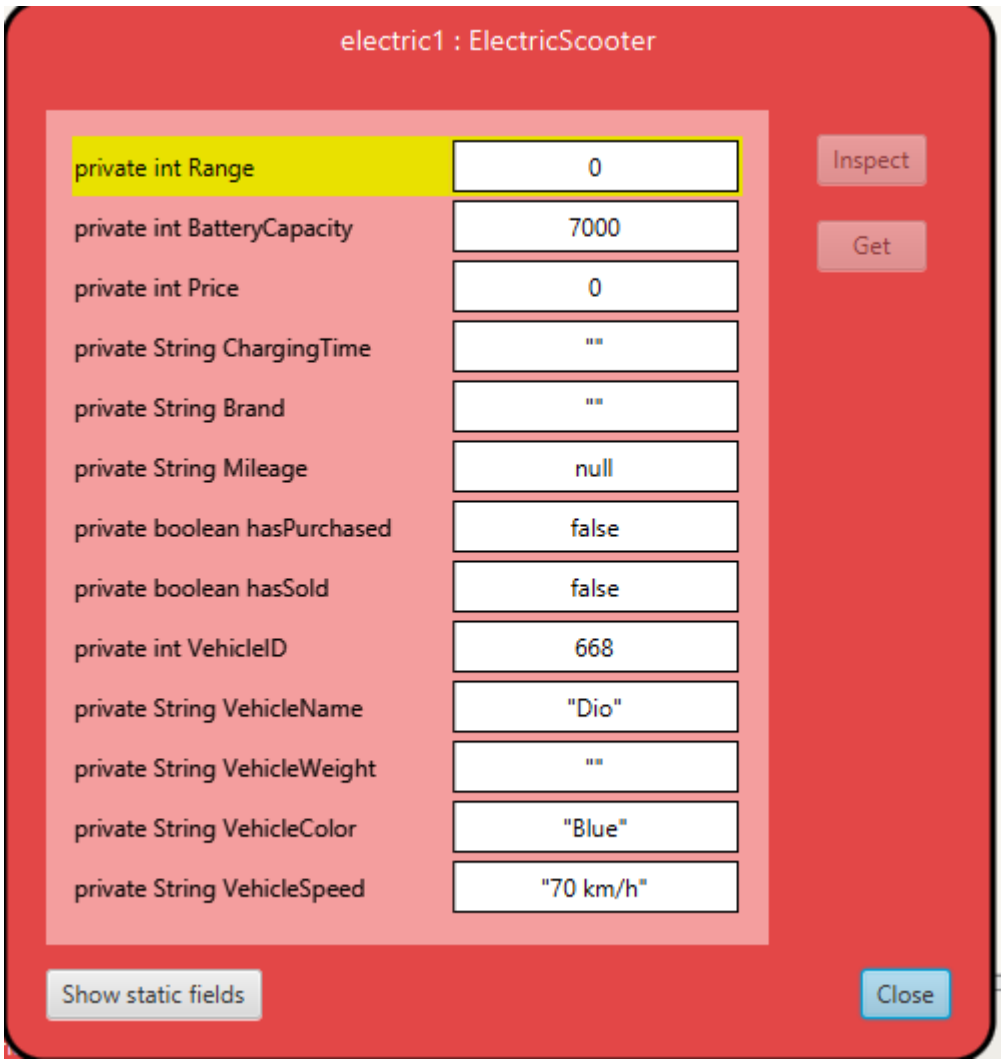

Figure 28: Inspecting ElectricScooter class

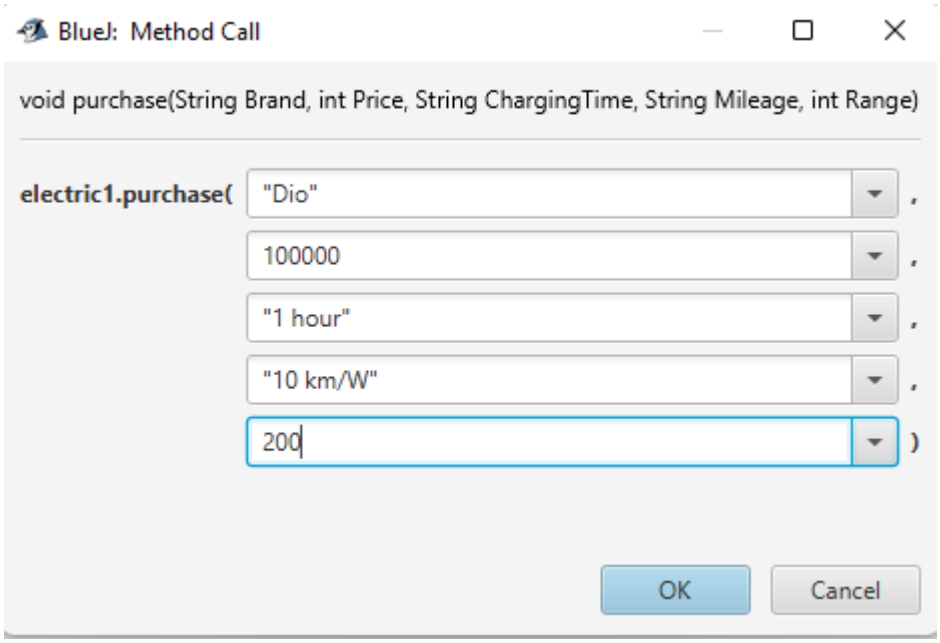

Figure 29: Purchasing Electric Scooter

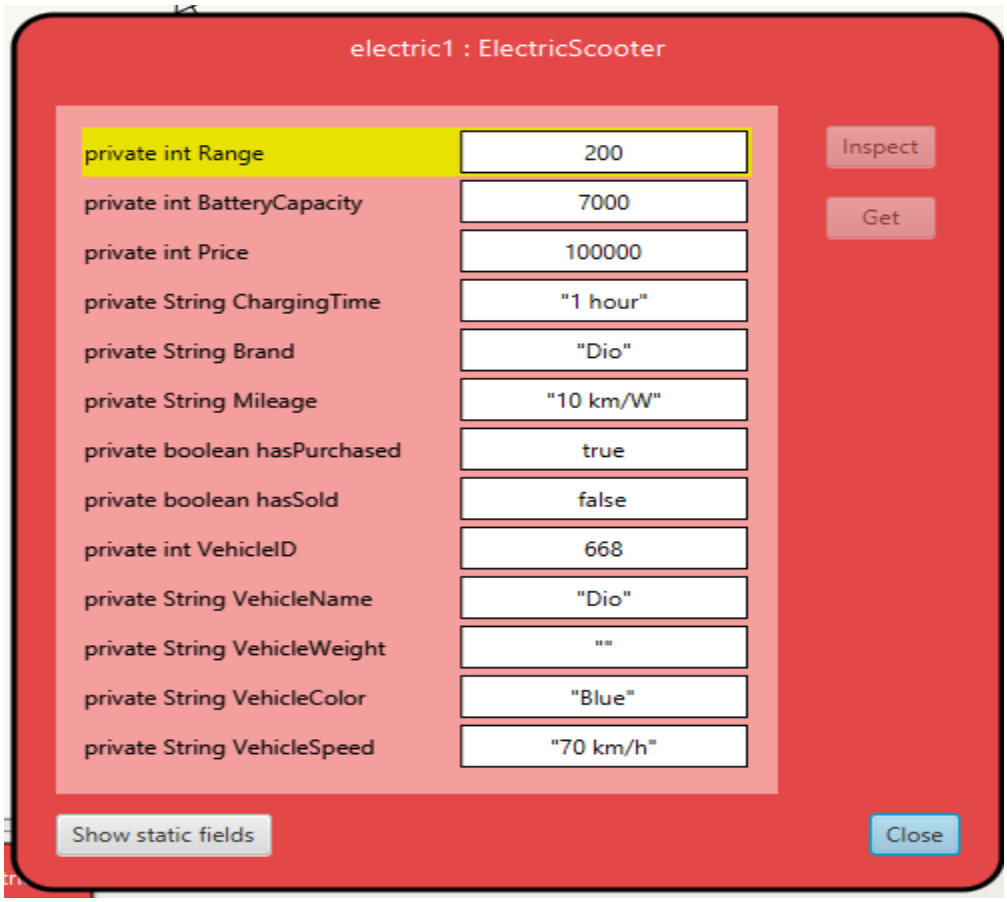

Figure 30: Re-Inspecting ElectricScooter Class

Objective	To purchase an Electric scooter
Action	Inspect, purchase, re-inspect
Expected Result	The electric scooter will be purchased.
Actual Result	The electric scooter has been purchased.
Conclusion	The test was successful.

Table 5: Test 2

5.3. Test 3

Figure 31: Inspecting ElectricScooter Class

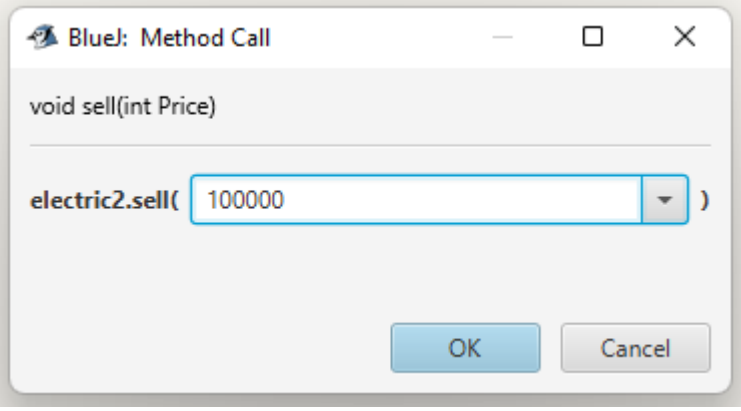

Figure 32: Selling Electric Scooter

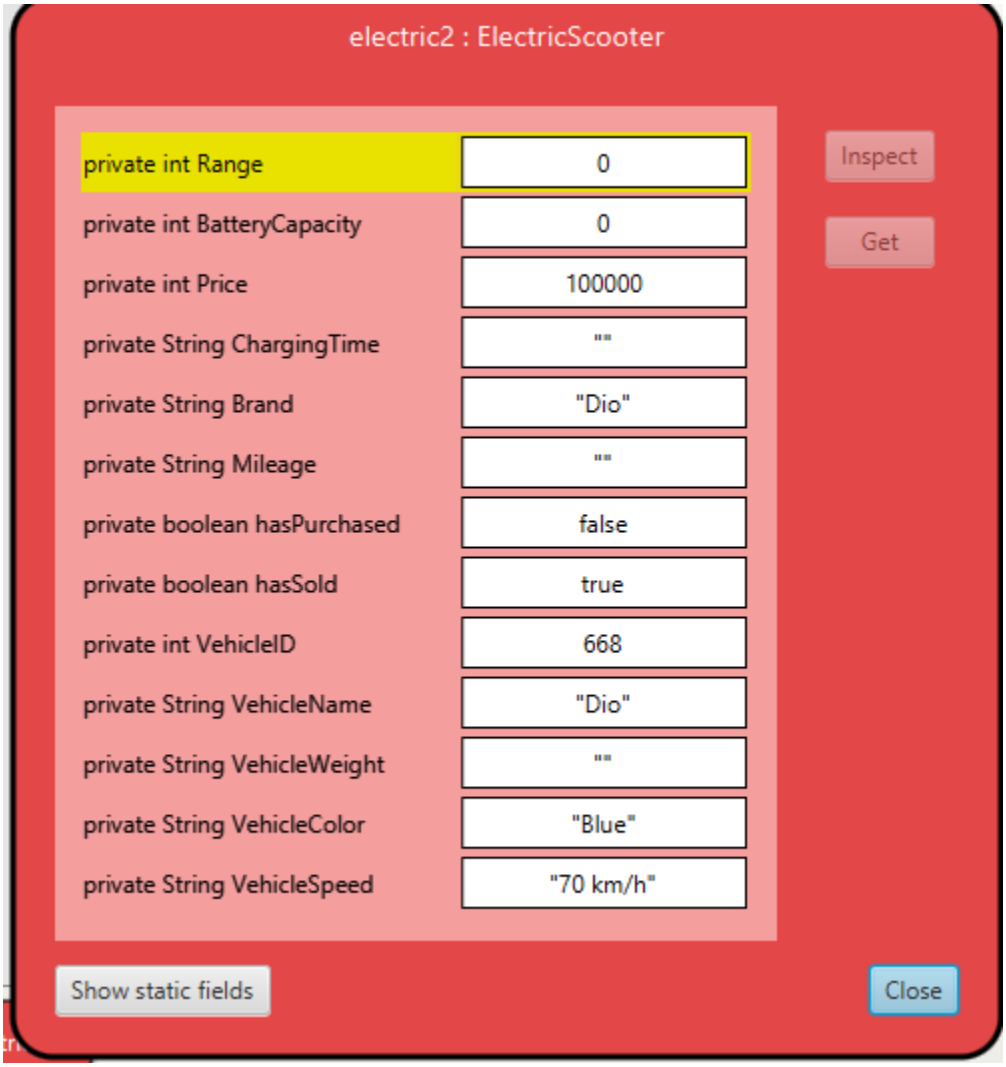

Figure 33: Re-Inspecting ElectricScooter Class

Objective	To change the status of hasPurchased to false.
Action	Inspect, sell, re-inspect.
Expected Result	The status of hasPurchased will be changed to false.
Actual Result	The status of hasPurchased has been changed to false
Conclusion	The test was successful.

Table 6: Test 3

5.4. Test 4

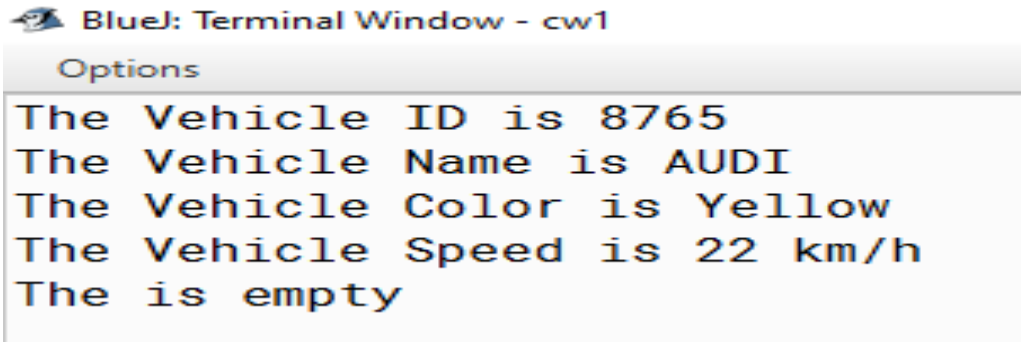

Figure 34: Detail of AutoRickshaw class

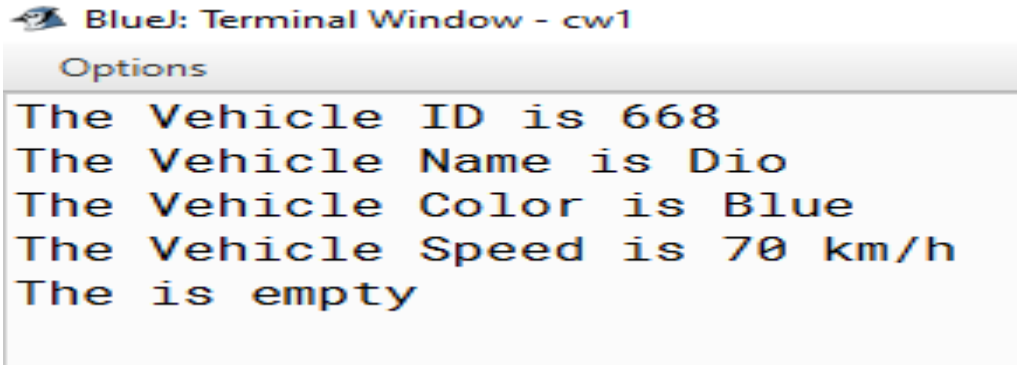

Figure 35: Detail of ElectricScooter class

Objective	To display Auto-Rickshaw and Electric Scooter details.
Action	Display.
Expected Result	The Auto-Rickshaw and Electric scooter will be displayed.
Actual Result	The Auto-Rickshaw and Electric scooter has been displayed.
Conclusion	The test was successful.

Table 7: Test 4

6. Error Detection

The errors that occurred during this coursework are listed below:

6.1. Error 1

When I complied with Vehicle class, in display method VehicleID wasn't displayed, and I looked into the code again and tried to fix the error.

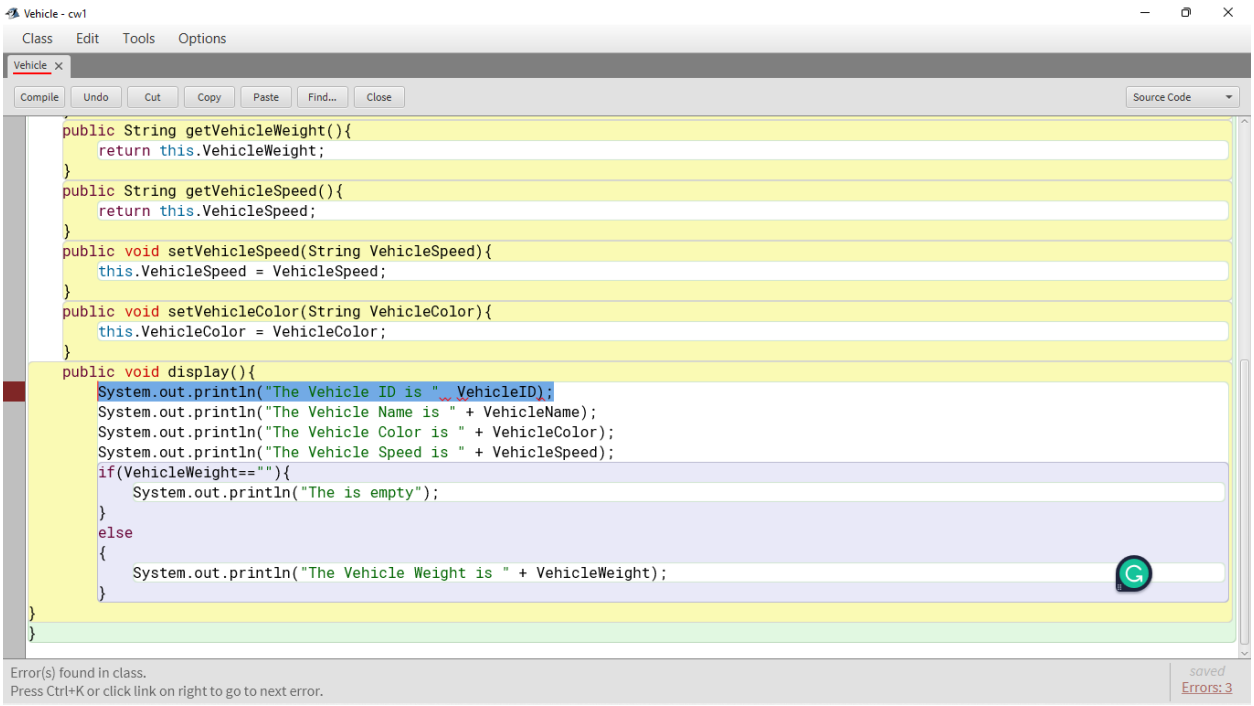

Figure 36: Error 1

6.2. Correction of error 1:

I fixed the missing concatenate sign (+) in the display of VehicleID.

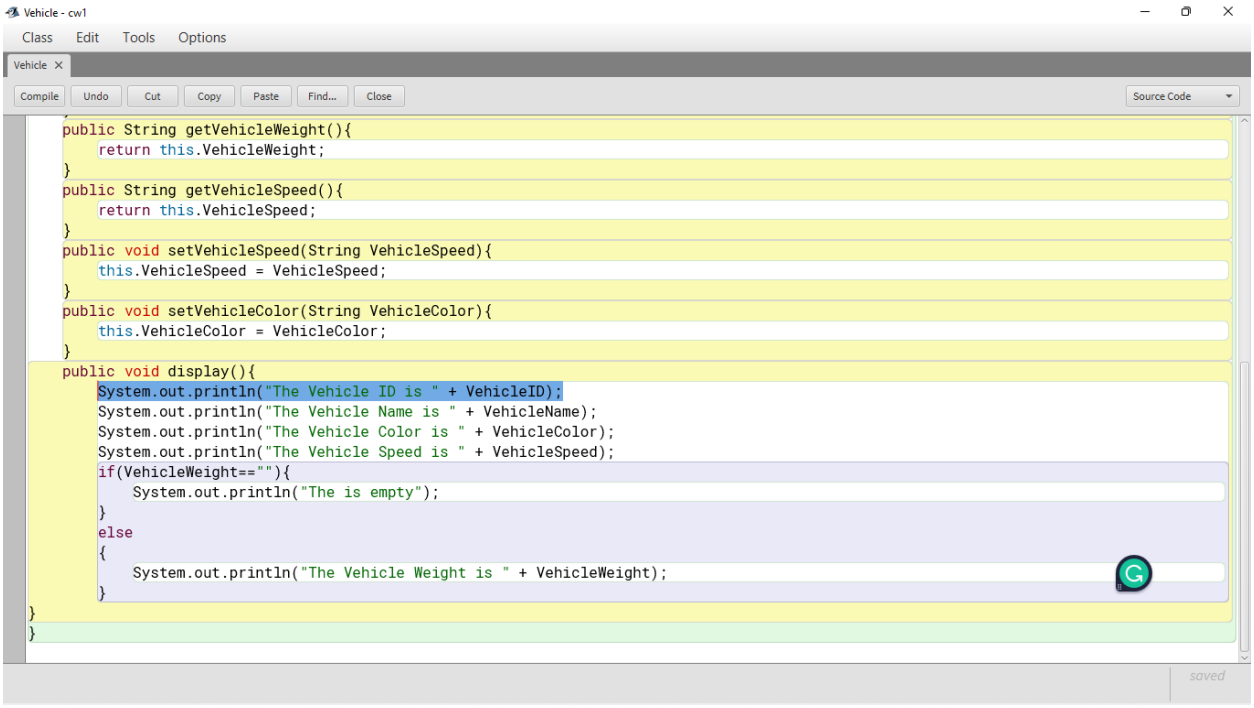

Figure 37: Correction of error 1

6.3. Error 2:

I mistakenly put wrong accessor method for EngineDisplacement. It was incompatible error which I tried to fix.

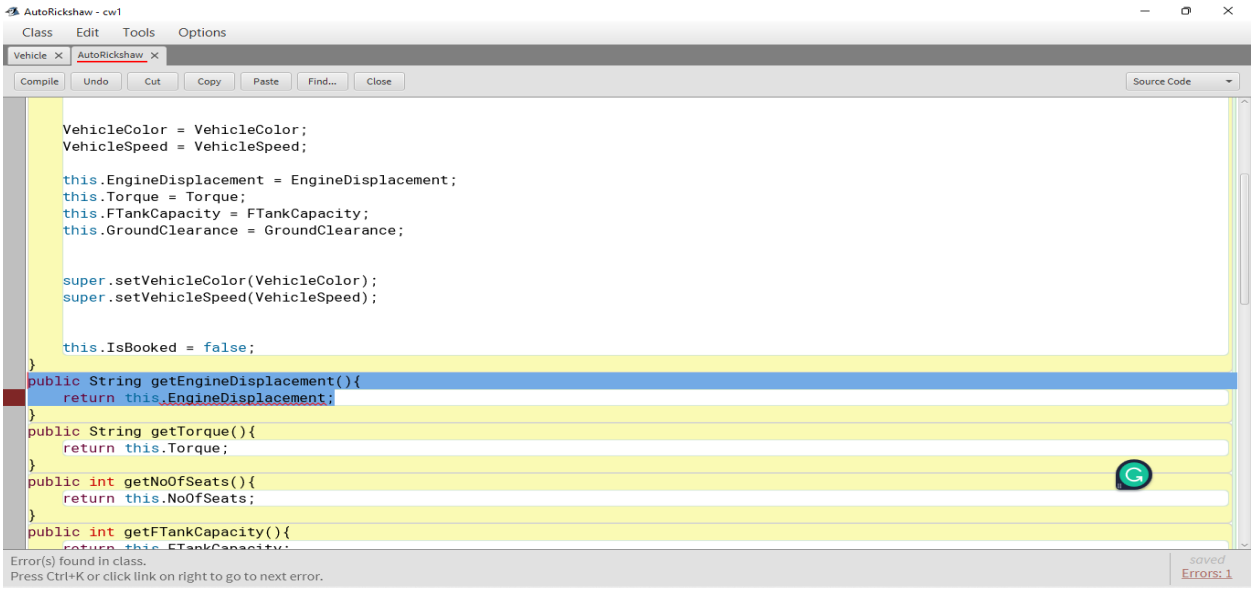

Figure 38: Error 2

6.4. Correction of error 2:

Finally, I figured it out that I used String data type so I corrected my error by setting int data type.

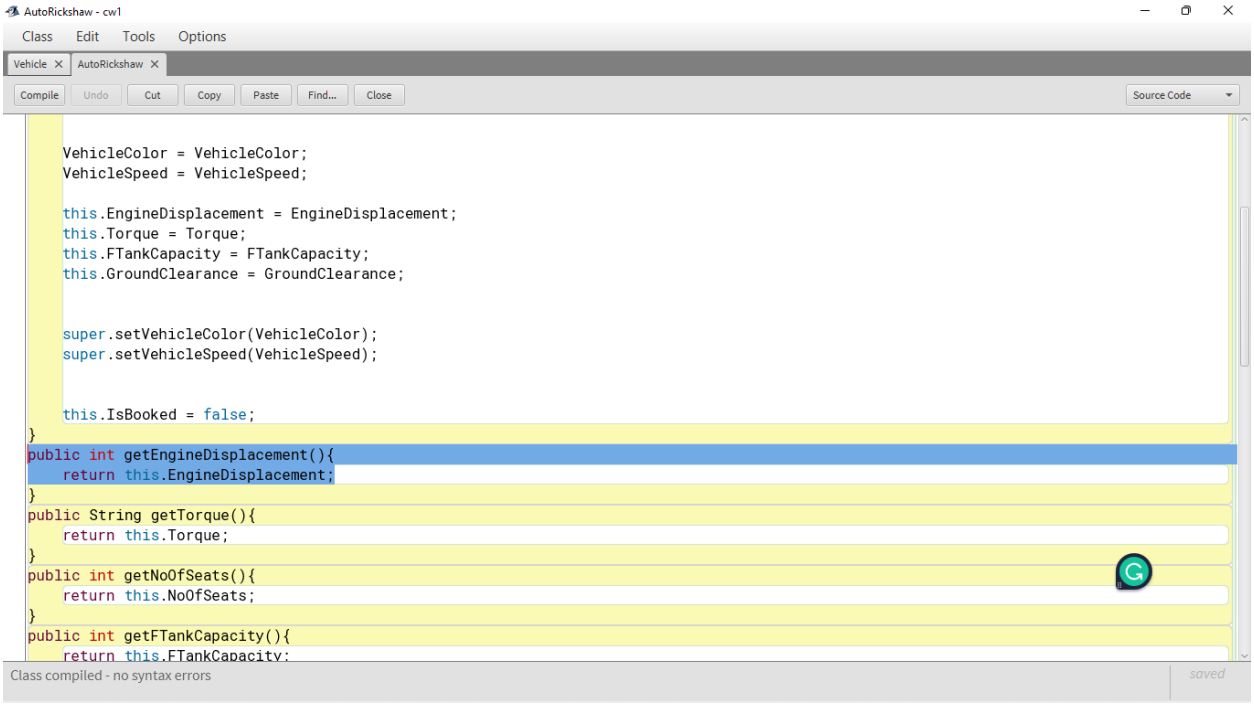

Figure 39: Correction of error 2

6.5. Error 3:

I faced the error which said, “class, interface or enum expected”. It took a long time for me to realize my mistake.

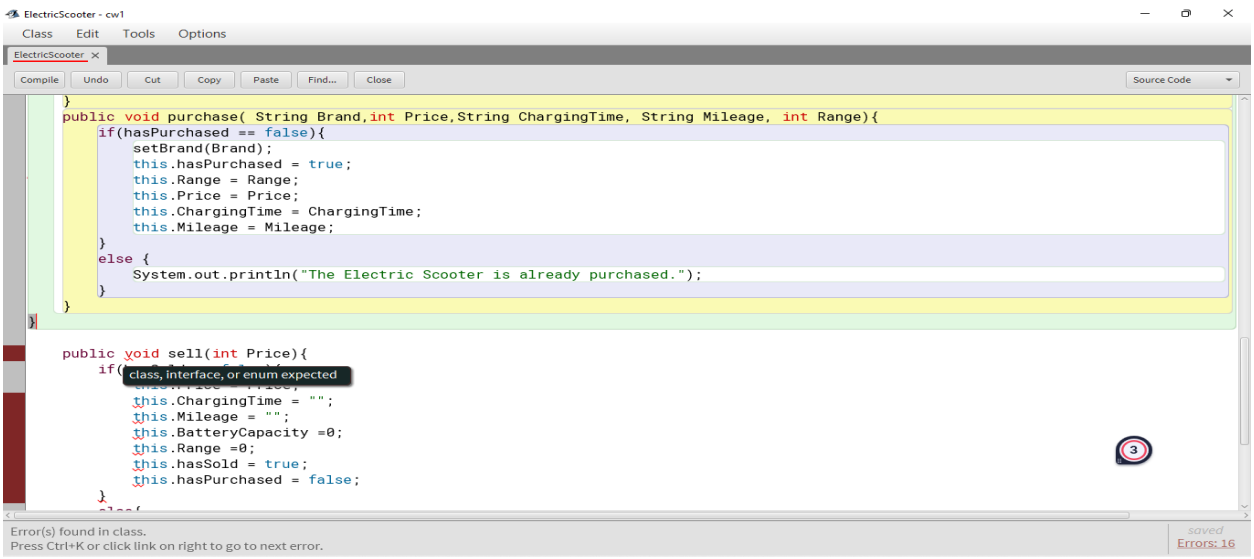

Figure 40: Error 3

6.6. Correction of error 3:

The error was I had put extra braces ahead of sell method. It was unnecessary. Excluding that unnecessary braces solved my error.

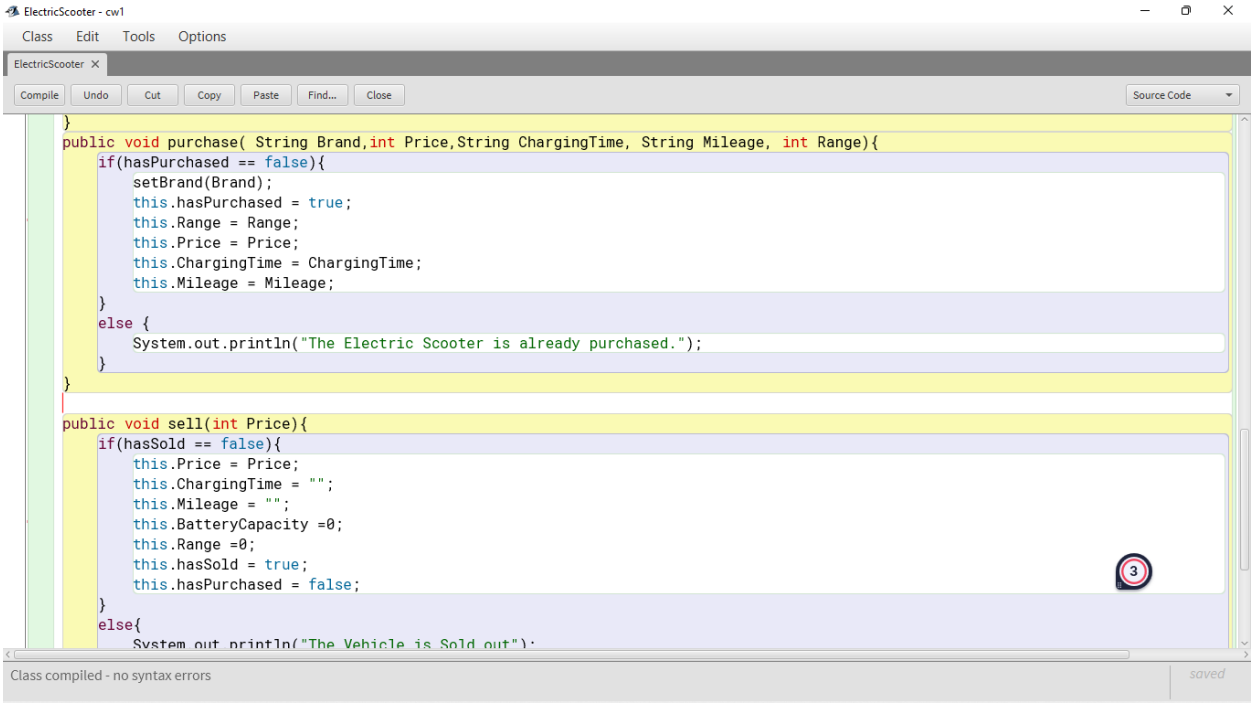

Figure 41: Correction of error 3

7. Conclusion:

This was our first coursework in this module. Working on this coursework was very fruitful in terms of earning and applying the knowledge I got from lectures, tutorials, and workshop classes. During the coding part I faced many problems as I got lots of error and figuring out those problems was the biggest headache for me. I learned that writing program needs creativity, patience, and researching skills to write best possible program. The biggest boost for me were the tutorials and workshop classes and mainly the discussion classes to complete this coursework under the given deadline.

I researched from different sources like websites, physical books, YouTube videos to get more knowledge. This coursework also enhanced my researching skills and I am so sure that it will contribute to make me better developer for my future. I was insecure about writing programs in Java before this coursework but after finishing this it helped me to overcome my insecurities. Time management and fear of not completing my coursework on time were the biggest challenges I faced during this time and now I am really glad that I am able to summit my coursework on time. Overall, it was the fun experience and I enjoyed doing this coursework.

Appendix:

a. Vehicle Class:

```
public class Vehicle{
   // Vehicel is the parent class
   private int VehicleID;      //VehicleID attribute in int data type
   private String VehicleName;     //VehicleName attribute in String data type
   private String VehicleWeight;   //VehicleWeight attribute in String data type
   private String VehicleColor;    //VehicleColor attribute in String data type
   private String VehicleSpeed;    //VehicleSpeed attribute in String data type

   public Vehicle( int VehicleID, String VehicleName, String VehicleColor,String VehicleWeight){
      // Vehicle constructor initializing VehicleID, VehicleName, VehicleColor, VehicleWeight in parameters
      this.VehicleID = VehicleID;
      this.VehicleName = VehicleName;
      this.VehicleColor = VehicleColor;
      this.VehicleWeight  =  "";
   }
   public int getVehicleID(){      //Crossponding accesor method for VehicleID
      return this.VehicleID;
   }
   public String getVehicleName(){     //Crossponding accesor method for VehicleName
      return this.VehicleName;
   }
   public String getVehicleColor(){       //Crossponding accesor method for VehicleColor
      return this.VehicleColor;
   }
   public String getVehicleWeight(){      //Crossponding accesor method for VehicleWeight
      return this.VehicleWeight;
```

```
    }
    public String getVehicleSpeed(){       //Crossponding accesor method for VehicleSpeed
        return this.VehicleSpeed;
    }
    public void setVehicleSpeed(String VehicleSpeed){          //setter method for
VehicleSpeed
        this.VehicleSpeed = VehicleSpeed;
    }
    public void setVehicleColor(String VehicleColor){       //setter method for Vehicle Color
        this.VehicleColor = VehicleColor;
    }
    public void display(){         //display method
        System.out.println("The Vehicle ID is " + VehicleID);
        System.out.println("The Vehicle Name is " + VehicleName);
        System.out.println("The Vehicle Color is " + VehicleColor);
        System.out.println("The Vehicle Speed is " + VehicleSpeed);
        if(VehicleWeight==""){
            System.out.println("The is empty");
        }
        else
        {
            System.out.println("The Vehicle Weight is " + VehicleWeight);
        }
}
}
```

b. AutoRickshaw class:

```
public class AutoRickshaw extends Vehicle{
    //subclass of Vehicle class
    private int EngineDisplacement;
    private String Torque;
    private int NoOfSeats;
    private int FTankCapacity;
    private String GroundClearance;        // attributes
    private int ChargeAmount;
    private String BookedDate;
    private boolean IsBooked;

    // values in parameters are initialized in AutoRickshaw constructor
public AutoRickshaw(int VehicleID, String VehicleName, String VehicleWeight, String
VehicleColor, String VehicleSpeed,
int EngineDisplacement, String Torque, int FTankCapacity, String GroundClearance){
    super(VehicleID, VehicleName, VehicleWeight, VehicleColor);


    VehicleColor = VehicleColor;
    VehicleSpeed = VehicleSpeed;

    this.EngineDisplacement = EngineDisplacement;
    this.Torque = Torque;
    this.FTankCapacity = FTankCapacity;
    this.GroundClearance = GroundClearance;


    super.setVehicleColor(VehicleColor);
    super.setVehicleSpeed(VehicleSpeed);
```

```
    this.IsBooked = false;
}
public int getEngineDisplacement(){              //Crossponding accesor method for
EngineDisplacement
    return this.EngineDisplacement;
}
public String getTorque(){      //Crossponding accesor method for Torque
    return this.Torque;
}
public int getNoOfSeats(){      //Crossponding accesor method for NoOfSeats
    return this.NoOfSeats;
}
public int getFTankCapacity(){      //Crossponding accesor method for Fuel tank capacity
    return this.FTankCapacity;
}
public String getGroundClearance(){              //Crossponding accesor method for
GroundClearance
    return this.GroundClearance;
}
public int getChargeAmount(){       //Crossponding accesor method for  chargeAmount
    return this.ChargeAmount;
}
public String getBookedDate(){      //Crossponding accesor method for BookedDate
    return this.BookedDate;
}
public boolean getIsBooked(){       //Crossponding accesor method for IsBooked
    return this.IsBooked;
}
public void setChargeAmount(int ChargeAmount){      //setter method  for ChargeAmount
    this.ChargeAmount = ChargeAmount;
```

```
}
public void setNoOfSeats(int NoOfSeats){        //setter method for NoOfSeats
    this.NoOfSeats = NoOfSeats;
}
public void BookAutoRickshaw(String BookedDate, int ChargeAmount, int NoOfSeats){
//Method to book an AutoRickshaw
    if(IsBooked == true){
        System.out.println("The Auto Rickshaw is booked");
        System.out.println("Status: " +IsBooked);
    }
    else{
        this.IsBooked = true;
        this.setChargeAmount(ChargeAmount);
        this.setNoOfSeats(NoOfSeats);
        this.BookedDate = BookedDate;
    }
}
public void Display(){      //Method to display
    super.display();        //Super keyword to call display method of Vehicle class
    if(IsBooked == true){
        System.out.println("The Engine Displacement is: " + EngineDisplacement);
        System.out.println("The torque is :" + Torque);
        System.out.println("The Number of Seats are :" + NoOfSeats);
        System.out.println("The total fuel tank capacity is :" + FTankCapacity);
        System.out.println("The Ground Clearance is :" + GroundClearance);
        System.out.println("The total charged amount is :" + ChargeAmount);
        System.out.println("The booking date is :" + BookedDate);
}
}
}
```

c. ElectricScooter:

```
public class ElectricScooter extends Vehicle{
    //sub class of Vehicle class
   private int Range;
   private int BatteryCapacity;
   private int Price;
   private String ChargingTime;     //attributes
   private String Brand;
   private String Mileage;
   private boolean hasPurchased;
   private boolean hasSold;

   //values in parameter are initialized  in ElectricScooter constructor
   public ElectricScooter(int VehicleID, String VehicleName, String VehicleWeight, String
VehicleSpeed, String VehicleColor, int BatteryCapacity){
       super(VehicleID, VehicleName, VehicleWeight, VehicleColor);

       super.setVehicleSpeed(VehicleSpeed);
       super.setVehicleColor(VehicleColor);



       this.Range = 0;
       this.BatteryCapacity = BatteryCapacity;
       this.Price = 0;
       this.ChargingTime = "";
       this.Brand = "";
       this.Mileage = Mileage;
       this.hasPurchased = false;
       this.hasSold = false;
    }
    public int getRange(){      //Crossponding accesor method for Range
```

```
        return this.Range;
    }
    public int getBatteryCapacity(){      //Crossponding accesor method for BatteryCapacity
        return this.BatteryCapacity;
    }
    public int getPrice(){      //Crossponding accesor method for Price
        return this.Price;
    }
    public String getChargingTime(){      //Crossponding accesor method for ChargingTime
        return this.ChargingTime;
    }
    public String getBrand(){       //Crossponding accesor method for Brand
        return this.Brand;
    }
    public String getMileage(){     //Crossponding accesor method for Mileage
        return this.Mileage;
    }
    public boolean getHasPurchased(){               //Crossponding accesor method for
HasPurchased
        return this.hasPurchased;
    }
    public boolean getHasSold(){        //Crossponding accesor method for HasSold
        return this.hasSold;
    }
    public void setBrand(String Brand){     //setter method for Brand
        if(hasPurchased != true){       //checking the value of Brand
            this.Brand = Brand;
        }
        else {
            System.out.println("Brand cannot be changed as it's already purchased");
        }
```

```
    }
    public void purchase( String Brand,int Price,String ChargingTime, String Mileage, int Range){   //method to purchase electric scooter
        if(hasPurchased == false){
            setBrand(Brand);
            this.hasPurchased = true;
            this.Range = Range;
            this.Price = Price;
            this.ChargingTime = ChargingTime;
            this.Mileage = Mileage;
        }
        else {
            System.out.println("The Electric Scooter is already purchased.");
        }
    }

    public void sell(int Price){        //method to sell the electric scooter
        if(hasSold == false){
            this.Price = Price;
            this.ChargingTime = "";
            this.Mileage = "";
            this.BatteryCapacity =0;
            this.Range =0;
            this.hasSold = true;
            this.hasPurchased = false;
        }
        else{
            System.out.println("The Vehicle is Sold out");
        }
    }
    public void DISPLAY(){      //method to display
```

```
        super.display();        //super keyword to call the display method of Vehicle class
        if(hasPurchased==false){
            System.out.println("The Electric Scooter brand is: " + Brand);
            System.out.println("The battery capacity of electric scooter is: " +
BatteryCapacity);
            System.out.println("The electric scooter gives the mileage of: " + Mileage);
            System.out.println("The range of electric scooter is: " + Range);
            System.out.println("The charging time of scooter is: " + ChargingTime);
        }
    }
}
```

Bibliography

(2022, March 03). Retrieved from Wekipedia: https://en.wikipedia.org/wiki/Diagrams.net

Unknown. (n.d.). *BlueJ.org/about*. Retrieved from BlueJ.org: https://www.bluej.org/about.html

Wikipedia . (2022, May 16). Retrieved from https://en.wikipedia.org/wiki/Microsoft_Word

WIkiPedia. (29, November 2018). Retrieved from https://en.wikipedia.org/wiki/Microsoft_Word#/media/File:Microsoft_Office_Word_(2019%E2%80%93present).svg