LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4001NI Programming

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2021 - 22 Autumn - 1

Student Name: Anish lamichhane

Group: computing c4

London Met ID: 21039633

College ID: NP01CP4A210056

Assignment Due Date: 22 Aprill 2022

Assignment Submission Date: 22 Aprill 2022

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 to the project

The graphical user interface is present in a wide range of the personal computers that we are using. Designing shapes and patterns is not like developing a graphical user interface. A realistic graphical user interface consists of a panels frame which your design, which features buttons, a text field, and other components. The target of this project is to add a class to the program you constructed for the first part of this assignment in order to build a (GUI) for a system that stores Fuelcar and Electriccar information in a list. A main method will be included in the class, and it will be validated to use the command prompt. This is the last coursework of first year, which had been given to us on week 202 and it is worth 20% of the overall grade.

Java is a class based high level object programing language. it is developed by ‘James gosling “and his friend in 1991. The purpose of the development of the java to make the java friendly with every operating system. The java is made for less implementation dependencies. It is the programing platform used for the application development. Java is the fast and secure programing language, it is easy to use and understand to different user. It is mainly used for developing java application in laptops, android and iOS device and other devices. The first java version is java jdk 1.0 and in 2010 java was officially bought by oracle company

GUI (Graphical User Interface) is a visual interface builder for Java programs that is simple to use. It is mostly made up of graphical elements such as buttons, labels, and windows that let the user to interact with a program**.** The main objective of this coursework is to developed a form to appoint a fuelcar or electric. The coursework defines about the hiring a staff who can work fuelcar or electriccar. GUI is created on the basis of three classes (Car, FuelCar, and Electric car) which is represented by another class GUI-DEMO. The frame with some of the attributes is created for the class GUI_DEMO using classes of swing package like JLabel. JTextField, JComboBox, JButton. And JLabel is used to display a short string in a frame (like CarID, CarnName CarPrice, Carcolor, CarPrice etc), JTextField is used for input or edit values. JComboBox is a popup menu

that shows a list for the users and here it is used to select FuelCar and Electriccar. JButton is used to create a labelled in this project we have JButton (like Add ,purchase , buy, Display and Clear).

BlueJ is a programming activity that enables you to quickly and easily create Java apps. BlueJ's designer has such a program called scope highlighting, that colors the backgrounds of each code to find things simpler to scan the text immediately. Also it aids inside the detecting of square brackets that have been lost. Learners will be ready to start working less quickly even without feeling confused as a consequence of this. BlueJ operates across Microsoft, Mac OS X, Linux, and other Programming language systems. It may also be run directly from the an Usb drive without need for installation.

2.Class diagram

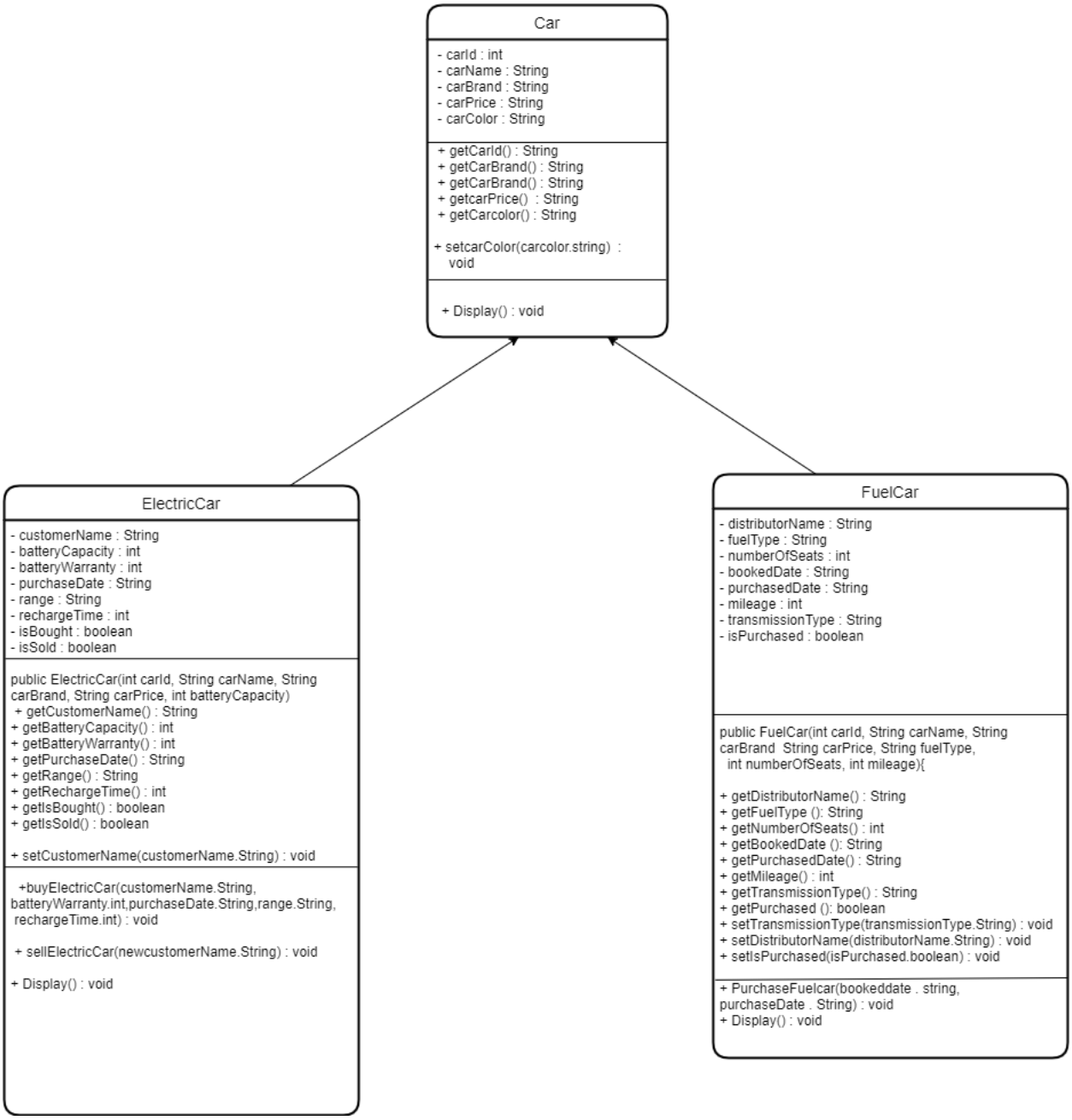


Figure 1. classdiagram of car

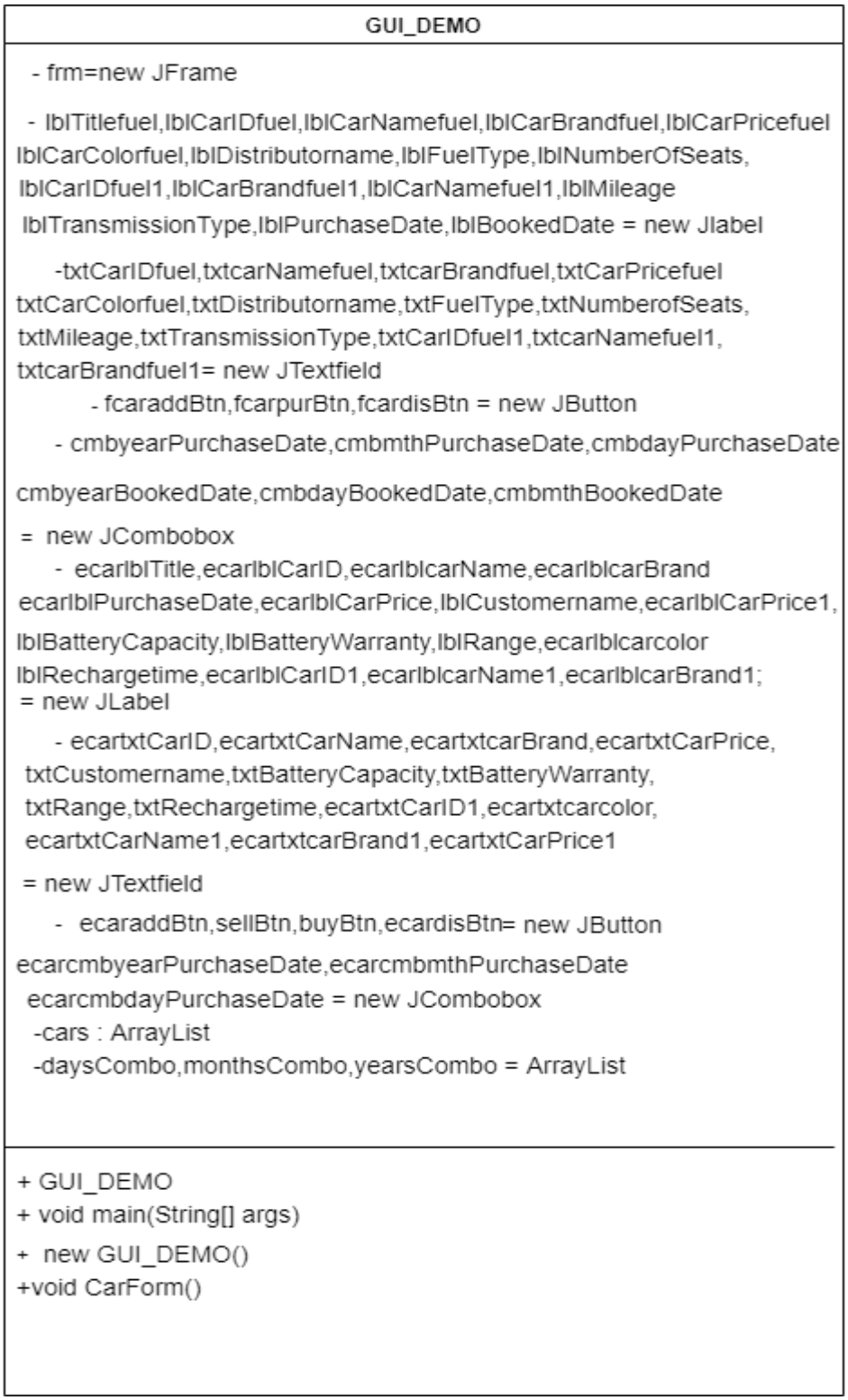


Figure 2class diagram of GUI_Demo

3. Method description

1. Void GUI

It is the instance method which consist GUI component like JFrame,JButton,JCombobox ,JLabel etc. the GUI layout is designed with the help of javax swing contain in the java file of blueJ. The layout of the component is GUI is designed by calling x axis and y axis and set their fix size in frame.

2. ActionPerformed

it is the method which is known as interface of the action Listener. the events of the parameter is accepted by GUI in this method, this method instantly call the registered components in GUI. The action is permuted by calling the add to Action Listener. when the action is performed the object contain in this method are appeal. It is used to add functionality to the button like add display etc.

3. Add method for Fuelcar

This method is performed by calling action listener to the add button of the FuelCar. the fuel car consist of carid, carname, carprice, carbrand,no of seats, fuel type and mileage. After filling all the components we click on the add button to call add method. the add method shows the car is added after adding valid data in fuelcar

4. Purchase method for Fuelcar

The value of CarID, Carprrice, carname ,transmission type, distributor name, purchased date, and booked date are entered in the GUI. When this button is pressed, the input value of CarD is compared to the existing CarID, and if valid CarD has been entered, it is used for the purchase of the fuel car in the list of car.

Display Method for fuelcar

After calling the purchase method and the value entered in the CarID, Carprrice, carname ,transmission type, distributor name, purchased date, and booked date and mileage. The information related to the Fuelcar is displayed in the console.

5. Add method for electric car

This method is performed by calling action listener to the add button of the ElectricCar. the Electric car consist of carid, carname, carprice, carbrand,no of seats, fuel type and Battery capacity . After filling all the components, we click on the add button to call add method. The ADD method shows the electric car is added after adding valid data in Electriccar

6. Buy method for electric car

The GUI is used to enter the carID, vehicle name, car brand, car price, car color, customer name, battery warranty, range, and recharge time. The purchase date's values are also entered. When the button is hit, the element of carID are compared with the previous carID, and if a correct carID is inputted, the electric car from either the list is purchased. The method for buying electric car .

7. Sell method for electric car

In the GUI, the carID & customer name are input. The input variable of a carID is compared with the previous carID in the list so when button is pressed. If you input a correct value, it can be used to sell a particular electric car from of the Car array list. This is the method for selling an electric car as from ElectricCar class.

8. Clear method

When this button is pressed in GUI, the values entered in the text fields are cleared from carform

4.PSEUDOCODE

PSEUDOCODE: Pseudo code is a non language specific description of what code should do. It allows the reader to understand what code they should implement with out tying it down to an implementation in one language. it's good for explanation.

Pseudocode

```
CREATE GUI-DEMO IMPLEMENTS ActionListener
FUNCTION main(String args[])
DO new GUI_DEMO().CarForm();
 END DO
END FUNCTION
FUNCTION car()
DO
        frm=new JFrame("Car");
        CALL new JLabel();
        CALL setText("Fuel car ");
        CALL setBounds(130, 20, 500, 40);
        CALL  frm.add(lblTitlefuel);
        CALL new Font("Serif",Font.BOLD,22);
         CALL setFont(ff);




      lblCarIDfuel=new JLabel("Car ID:");
      CALL lblCarIDfuel.  setBounds(20,70,100,20);
       CALL  frm.add(lblCarIDfuel);
```

```
txtCarIDfuel=new JTextField("");
CALL   lblcarNamefuel setBounds(90,70,100,20);
 CALL frm. add(txtCarIDfuel);

lblcarNamefuel=new JLabel("Car Name:");
CALL  lblcarNamefuel.setBounds(220, 70, 100, 20);
CALL frm.add(lblcarNamefuel);

txtcarNamefuel=new JTextField("");
CALL txtcarNamefuel setBounds(290,70,100,20);
CALL frm.add(txtcarNamefuel);

lblcarBrandfuel=new JLabel("Car Brand:");
CALL lblcarBrandfuel  setBounds(20,130,150,20);
CALL. Frm. add(lblcarBrandfuel);

txtcarBrandfuel=new JTextField("");
CALL. txtcarBrandfuel setBounds(90,130,100,20);
CALL frm. add(txtcarBrandfuel);

 LblCarPricefuel = new JLabel("Car Price:");
CALL lblCarPricefuel setBounds(220,130,150,20);
CALL frm.add (lblCarPricefuel);

txtCarPricefuel=new JTextField("");
CALL CarPricefuel setBounds(290,130,100,20);
CALL frm.add (txtCarPricefuel);

lblFuelType=new JLabel("Fuel Type:");
CALL lblFuelType.setBounds(20,190,200,20);
```

```
CALL frm.add (lblFuelType);

txtFuelType=new JTextField("");
CALL  txtFuelType.setBounds(90,190,100,20);
CALL frm.add(txtFuelType);

lblNumberofSeats=new JLabel("No of Seats");
CALL lblNumberofSeats.setBounds(220,190,200,20);
 CALL  frm.add(lblNumberofSeats);



txtNumberofSeats=new JTextField("");
 CALL txtNumberofSeats.setBounds(290,190,100,20);
CALL frm.add( txtNumberofSeats);


 lblMileage=new JLabel("Mileage:");
CALL lblMileage.setBounds(20,250,250,20);
 CALL frm.add(lblMileage);

txtMileage=new JTextField("");
CALL txtMileage.setBounds(90,250,100,20);
CALL  frm.add(txtMileage);

fcaraddBtn = new JButton("Add");
SET THE BACKGROUND fcaraddBtn to .BLUE);
CALL fcaraddBtn.setBounds(220,250,100,30);
CALL  frm.add(fcaraddBtn);
CALL  fcaraddBtn.addActionListener(this);
```

```
lblCarIDfuel=new JLabel("Car ID:");
CALL lblCarIDfuel.setBounds(20,300,300,20);
CALL frm.add(lblCarIDfuel);

txtCarIDfuel=new JTextField("");
CALL  txtCarIDfuel.setBounds(90,300,100,20);
CALL  frm.add(txtCarIDfuel);

lblcarNamefuel=new JLabel("Car Name:");
CALL  lblcarNamefuel.setBounds(220, 300, 300, 20);
CALL  frm.add(lblcarNamefuel);

txtcarNamefuel=new JTextField("");
CALL txtcarNamefuel.setBounds(290,300,100,20);
CALL  frm.add(txtcarNamefuel);

lblcarBrandfuel=new JLabel("Car Brand:");
CALL   lblcarBrandfuel.setBounds(20,360,300,20);
CALL frm.add(lblcarBrandfuel);

txtcarBrandfuel=new JTextField("");
CALL  txtcarBrandfuel.setBounds(90,360,100,20);
CALL  frm.add(txtcarBrandfuel);

lblCarColorfuel=new JLabel("Car Color:");
CALL  lblCarColorfuel.setBounds(220,360,300,20);
CALL frm.add(lblCarColorfuel);

txtCarColorfuel=new JTextField("");
```

```
CALL  txtCarColorfuel.setBounds(290,360,100,20);
CALL  frm.add(txtCarColorfuel);

lblTransmissionType=new JLabel("trans type:");
CALL  lblTransmissionType.setBounds(20,420,100,20);
CALL  frm.add(lblTransmissionType);

txtTransmissionType=new JTextField("");
CALL  txtTransmissionType.setBounds(90,420,100,20);
CALL  frm.add(txtTransmissionType);

lblFuelType=new JLabel("Fuel type:");
CALL  lblFuelType.setBounds(220,420,100,20);
CALL  frm.add(lblFuelType);

txtFuelType=new JTextField("");
CALL txtFuelType.setBounds(290,420,100,20);
CALL frm.add(txtFuelType);

lblPurchaseDate=new JLabel("Purchase Date");
CALL lblPurchaseDate.setBounds(20,460,100,20);
CALL frm.add(lblPurchaseDate);

String[] puryear = { "2000", "2001", "2002", "2003", "2004", "2005",
"2006","2007","2008", "2009", "2010", "2011","2012",
"2013", "2014", "2015", "2016", "2017" ,"2018", "2019", "2020", "2021","2022"};
cmbyearPurchaseDate=new JComboBox(puryear);
CALL  cmbyearPurchaseDate.setBounds(140,460,60,20);
CALL  frm.add(cmbyearPurchaseDate);
```

```
String[] purmonth = {"January","February", "March", "April", "May", "June","July",
"August", "September",
    "October", "November", "December"};
cmbmthPurchaseDate=new JComboBox(purmonth);
    CALL cmbmthPurchaseDate.setBounds(220,460,60,20);
    CALL     frm.add(cmbmthPurchaseDate);

String[] purday = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14"
,"15", "16", "17", "18", "19",
"20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30","31"};
cmbdayPurchaseDate=new JComboBox(purday);
  CALL cmbdayPurchaseDate.setBounds(300,460,60,20);
  CALL frm.add(cmbdayPurchaseDate);

lblBookedDate=new JLabel("Booked Date");
  CALL lblBookedDate.setBounds(20,510,100,20);
  CALL  frm.add(lblBookedDate);

String[] bookyear = { "2000", "2001", "2002", "2003", "2004", "2005",
"2006","2007","2008", "2009", "2010", "2011","2012",
"2013", "2014", "2015", "2016", "2017" ,"2018", "2019", "2020", "2021","2022"};
cmbyearBookedDate=new JComboBox(bookyear);
  CALL cmbyearBookedDate.setBounds(140,510,60,20);
  CALL  frm.add(cmbyearBookedDate);

String[] bookmonth = {"January","February", "March", "April", "May", "June","July",
"August", "September",
    "October", "November", "December"};
cmbmthBookedDate=new JComboBox(bookmonth);
  CALL cmbmthBookedDate.setBounds(220,510,60,20);
```

```
        CALL  frm.add(cmbmthBookedDate);

      String[] bookday = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14"
,"15", "16", "17", "18", "19",
      "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30","31"};
      cmbdayBookedDate=new JComboBox(bookday);
        CALL cmbdayBookedDate.setBounds(300,510,60,20);
        CALL  frm.add(cmbdayBookedDate);


     fcarpurBtn = new JButton("Purchase");
        SET THE BACKGROUND  fcarpurBtn.to yellow);
        CALL fcarpurBtn.setBounds(180,540,100,30);
        CALL  frm.add(fcarpurBtn);
      CALL fcarpurBtn.addActionListener(this);

     fcardisBtn = new JButton("Display");
         SET THE BACKGROUND TO  yellow);
        CALL fcardisBtn.setBounds(300,540,100,30);
        CALL frm.add(fcardisBtn);
   CALL fcardisBtn.addActionListener(this);


     ecarlblTitle =new JLabel();
         CALL ecarlblTitle.setText("Electric car ");
        CALL ecarlblTitle.setBounds(600, 20, 500, 40);
        CALL frm.add(ecarlblTitle);
        CALL  Font fff=new Font("Serif",Font.BOLD,22);
        CALL   ecarlblTitle.setFont(fff);
```

```
ecarlblCarID=new JLabel("Car ID:");
   CALL ecarlblCarID.setBounds(460,70,100,20);
   CALL frm.add(ecarlblCarID);

ecartxtCarID=new JTextField("");
   CALL ecartxtCarID.setBounds(530,70,100,20);
   CALL frm.add(ecartxtCarID);

ecarlblcarName=new JLabel("Car Name:");
   CALL ecarlblcarName.setBounds(660, 70, 100, 20);
   CALL frm.add(ecarlblcarName);

ecartxtcarName=new JTextField("");
   CALL   ecartxtcarName.setBounds(730,70,100,20);
   CALL frm.add(ecartxtcarName);

ecarlblcarBrand=new JLabel("Car Brand:");
   CALL ecarlblcarBrand.setBounds(460,130,150,20);
   CALL frm.add(ecarlblcarBrand);

ecartxtcarBrand=new JTextField("");
   CALL  ecartxtcarBrand.setBounds(530,130,100,20);
   CALL frm.add(ecartxtcarBrand);

ecarlblCarPrice=new JLabel("Car Price:");
   CALL ecarlblCarPrice.setBounds(660,130,150,20);
   CALL frm.add(ecarlblCarPrice);

ecartxtCarPrice=new JTextField("");
   CALL ecartxtCarPrice.setBounds(730,130,100,20);
```

```
    CALL frm.add(ecartxtCarPrice);

  lblBatteryCapacity=new JLabel("Battery Capacity:");
    CALL lblBatteryCapacity.setBounds(460,190,150,20);
    CALL frm.add(lblBatteryCapacity);

  txtBatteryCapacity=new JTextField("");
    CALL txtBatteryCapacity.setBounds(570,190,100,20);
    CALL frm.add(txtBatteryCapacity);

  ecaraddBtn = new JButton("Add");
    SET THE BACKGROUND OF  ecaraddBtn to  red);
    CALL ecaraddBtn.setBounds(730,190,100,30);
    CALL frm.add(ecaraddBtn);
    CALL ecaraddBtn.addActionListener(this);


  ecarlblCarID=new JLabel("Car ID:");
    CALL ecarlblCarID.setBounds(460,240,100,20);
    CALL  frm.add(ecarlblCarID);

  ecartxtCarID=new JTextField("");
 CALL ecartxtCarID.setBounds(530,240,100,20);
 CALL frm.add(ecartxtCarID);

  ecarlblcarName=new JLabel("Car Name:");
  CALL ecarlblcarName.setBounds(660, 240, 100, 20);
  CALL frm.add(ecarlblcarName);

  ecartxtcarName=new JTextField("");
```

```
CALL ecartxtcarName.setBounds(730,240,100,20);
CALL frm.add(ecartxtcarName);

 ecarlblcarBrand=new JLabel("Car Brand:");
CALL ecarlblcarBrand.setBounds(460,300,150,20);
CALL frm.add(ecarlblcarBrand);

 ecartxtcarBrand=new JTextField("");
CALL ecartxtcarBrand.setBounds(530,300,100,20);
CALL frm.add(ecartxtcarBrand);

 ecarlblCarPrice=new JLabel("Car Price:");
CALL ecarlblCarPrice.setBounds(660,300,150,20);
 CALL frm.add(ecarlblCarPrice);

 ecartxtCarPrice=new JTextField("");
CALL ecartxtCarPrice.setBounds(730,300,100,20);
CALL  frm.add(ecartxtCarPrice);

 ecarlblcarBrand=new JLabel("Car Color:");
CALL ecarlblcarBrand.setBounds(460,360,150,20);
CALL frm.add(ecarlblcarBrand);

 ecartxtcarBrand=new JTextField("");
CALL ecartxtcarBrand.setBounds(530,360,100,20);
 CALL frm.add ecartxtcarBrand

 lblBatteryCapacity=new JLabel("Warranty:");
   CALL lblBatteryCapacity.setBounds(660,360,150,20);
   CALL frm.add(lblBatteryCapacity);
```

```
 txtBatteryCapacity=new JTextField("");
CALL txtBatteryCapacity.setBounds(730,360,100,20);
CALL frm.add(txtBatteryCapacity);

 lblCustomername=new JLabel("Cust Name:");
CALL lblCustomername.setBounds(460,420,150,20);
CALL  frm.add(lblCustomername);

 txtCustomername=new JTextField("");
CALL  txtCustomername.setBounds(530,420,100,20);
CALL frm.add(txtCustomername);

 lblRange=new JLabel("Range:");
CALL  lblRange.setBounds(660,420,150,20);
CALL  frm.add(lblRange);

 txtRange=new JTextField("");
 CALL txtRange.setBounds(660,420,150,20);
 CALL frm.add(txtRange);

 lblBatteryCapacity=new JLabel("Battery Capacity:");
 CALL lblBatteryCapacity.setBounds(460,480,150,20);
  CALL frm.add(lblBatteryCapacity);

 txtBatteryCapacity=new JTextField("");
CALL   txtBatteryCapacity.setBounds(560,480,100,20);
CALL   frm.add(txtBatteryCapacity);

 ecarlblPurchaseDate=new JLabel("Purchase Date");
```

```
CALL  ecarlblPurchaseDate.setBounds(460,530,100,20);
CALL  frm.add(ecarlblPurchaseDate);

 String[] purcyear = { "2000", "2001", "2002", "2003", "2004", "2005",
"2006","2007","2008", "2009", "2010", "2011","2012",
    "2013", "2014", "2015", "2016", "2017" ,"2018", "2019", "2020", "2021","2022"};
 ecarcmbyearPurchaseDate=new JComboBox(purcyear);
CALL ecarcmbyearPurchaseDate.setBounds(560,530,60,20);
CALL frm.add(ecarcmbyearPurchaseDate);

 String[] purcmonth = {"January","February", "March", "April", "May", "June","July",
"August", "September",
    "October", "November", "December"};
 ecarcmbmthPurchaseDate=new JComboBox(purcmonth);
CALL ecarcmbmthPurchaseDate.setBounds(640,530,60,20);
CALL frm.add(ecarcmbmthPurchaseDate);

 String[] purcday = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14"
,"15", "16", "17", "18", "19",
 "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30","31"};
 ecarcmbdayPurchaseDate=new JComboBox(purcday);
CALL ecarcmbdayPurchaseDate.setBounds(720,530,60,20);
CALL  frm.add(ecarcmbdayPurchaseDate);


buyBtn = new JButton("Buy");
SET THE BACKGROUND OF buyBtn.setBackground(Color.green);
CALL   buyBtn.setBounds(580,580,100,30);
CALL   frm.add(buyBtn);
CALL buyBtn.addActionListener(this);
```

```
      ecardisBtn = new JButton("Display");
       SET THE BACKGROUND OF  ecardisBtn to green);
     CALL  ecardisBtn.setBounds(460,580,100,30);
    CALL   frm.add(ecardisBtn);
CALL ecardis.addActionListener(this);

      sellBtn = new JButton("Sell");
     SET THE BACKGROUND OF sellBtn to .green):
      CALL sellBtn.setBounds(710,580,100,30);
       CALL    frm.add(sellBtn);
       CALL sellBtn.addActionListener(this);

      clrBtn = new JButton("Clear");
    SET THE BACKGROUND OF  clrBtn to setBackground(Color.red);
     CALL  clrBtn.setBounds(350,600,100,30);
      CALL frm.add(clrBtn);
CALL CltBtn.addActionListener(this);



     CALL   frm.setSize(900,700);
    CALL    frm.setLayout(null);
     CALL frm.setResizable(false);
    CALL    frm.setVisible(true);
    CALL     frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
     END DO
      END FUNCTION
 FUNCTION actionPerformed(ActionEvente)
  DO IF(e.getSource()==fcaraddBtn)
```

```
 LET NUMBER carId=0;
LET NUMBER numberOfSeats=0;
LET NUMBER mileage =0;
LET STRING carName="";
 LET STRING carBrand ="";
   LET STRING    fuelType="";
    LET STRING carPrice
TRY DO
carId = CONVERT INTO NUMBER(txtCarIDfuel.getText());
                carName = txtcarNamefuel.getText();
                carBrand = txtcarBrandfuel.getText();
                fuelType = txtFuelType.getText();
                numberOfSeats = convert into number (txtNumberofSeats.getText());
                mileage = convert into number (txtMileage.getText());
                carPrice = txtCarPricefuel.getText();
               SET boolean isDuplicatecarId  to false;

                FOR(Car cars:list){
                    IF(cars.getCarId() == carId){
                       Set  isDuplicatecarId to true;
                        break;
                    END IF
                END  FOR

                IF (isDuplicatecarId  is equals to the false){

FuelCarfuel=newFuelCar(carId,carName,carBrand,carPrice,fuelType,numberOfSeats,m
ileage);
                list.add(fuel);
                SHOWDIALOGBOX(frm,"Car has been added."+list.size());
```

```
            END IF
           ELSE
            SHOWDIALOGBOX(frm," Car Id already exits. Please enter the new Car Id”);
      END ELSE
      END DO
      END TRY

CATCH(Exception exp1)
SHOWDIALOGBOX("please enter valid info")
 END CATCH
END IF
IF
txtCarIDfuel.setText("");
          txtcarNamefuel.setText("");
          txtcarBrandfuel.setText("");
          txtCarPricefuel.setText("");
          txtCarColorfuel.setText("");
          txtFuelType.setText("");
          txtNumberofSeats.setText("");
          txtMileage.setText("");
          txtTransmissionType.setText("");
          txtCarIDfuel1.setText("");
          txtcarNamefuel1.setText("");
          txtcarBrandfuel1.setText("");
          txtDistributorname.setText("");
          // clearing textfield of Electriccar
          ecartxtCarID.setText("");
          ecartxtCarName.setText("");
          ecartxtcarBrand.setText("");
          ecartxtCarPrice.setText("");
```

```
        txtBatteryCapacity.setText("");
        txtBatteryWarranty.setText("");
        txtCustomername.setText("");
        txtRange.setText("");
        ecartxtcarcolor.setText("");
        ecartxtCarID1.setText("");
        ecartxtCarName1.setText("");
        ecartxtcarBrand1.setText("");
        ecartxtCarPrice1.setText("");
        txtRechargetime.setText("");
END IF
END FUNCTION
IF
If
 DO IF (e.getSource() == ecaraddBtn){
            LET NUMBER carId2= 0;
           LET STRING carName = "";
           LET STRING carBrand = "";
           LET NUMBER BatteryCapacity = 0;
           LET STRING carPrice = "";
           TRY
           DO

{
                int carID = CONVERT TO NUMBER(ecartxtCarID.getText());
                String CarName = ecartxtCarID.getText();
                String CarBrand = ecartxtcarBrand.getText();
                String CarPrice = ecartxtCarPrice.getText();
                intbatteryCapacity= CONVERT TO NUMBER(txtBatteryCapacity.getText());
               SET  isDuplicatecarId  TO false;
```

```
            FOR(Car cars:list){
               SET (cars.getCarId() is  equals to carId2){
                 SET  isDuplicatecarId to true;
                  END IF
                 END FOR
            IF (isDuplicatecarId is equals false){

ElectricCarelectric=newElectricCar(carId2,carName,carBrand,carPrice,batteryCapacity);
            list.add(electric);
            SHOWDIALOGBOX(,"Car has been added."+list.size());
            END IF
           ELSE
               SHOWDIALOGBOX ("Car Id already exits. Please enter the new Car Id");
             END ELSE

          CATCH(expection ex1)
           SHOW DIALOG(“"Please check if you have entered valid information”)
         END CATCH
         END TRY
IF (e.getSource()==ecardisBtn)

         IF(list.size()<=0){
         SHOWDIALOG  (,"Electric Car is not Added");
        END IF
        ELSE
             SHOWDIALOG ("Electric Car detail is displayed");
           FOR(Car var5:list){
              IF(var5 instanceof ElectricCar){
                 DISPLAY ("Electric car is displayed: ");
```

```
                ElectricCar obj5=(ElectricCar)var5;
               CALL obj5.display();
              SHOWDIALOG  (frm,"Electric Car detail is displayed");
          END IF
        END FOR
        END DO



            IF(e.getSource()==fcardisBtn)

        SHOW DIALOG(frm,"FuelCar Details is displayed");
        IF (list.size()<=0){
            SHOW DIALOG (frm,"car is not added");
        END IF
        ELSE
            FOR (Car var5:list){
                IF(var5 instanceof FuelCar){
                    DISPLAY("fuelcar is displayed: ");
                    FuelCar fuel=(FuelCar) var5;
                   CALL  fuel.display();
               END IF
               END FOR
                END ELSE
                  END IF


IF (e.getSource() == fcarpurBtn){
        TRY
        DO
```

```
        INT carId = CONVERT INTO NUMBER(txtCarIDfuel1.getText());
        STRING carName = txtcarNamefuel1.getText();
        STRING carBrand = txtcarBrandfuel1.getText();
        STRING carColor = txtCarColorfuel.getText();
        STRING transmissionType = txtTransmissionType.getText();
        STRING distributorName = txtDistributorname.getText();

StringpurchasedDate=CONVERTINTOSTRING(cmbyearPurchaseDate.getSelectedIte
m())+CONVERTINTOSTRING(cmbmthPurchaseDate.getSelectedItem())+
CONVERTINTOSTRING (cmbdayPurchaseDate.getSelectedItem
        StringbookedDate=CONVERTINTOSTRING
(cmbyearBookedDate.getSelectedItem())+CONVERTINTOSTRING
(cmbmthBookedDate.getSelectedItem()+CONVERTINTOSTRING(cmbdayBookedDate.
getSelectedItem
        SET boolean isCarIdThere  TO false;
        FOR(Car car:list){
           IF(car.getCarId() == carId)
              IF(car instanceof FuelCar)
                 FuelCar fuelCar  IS(FuelCar)car
                 if(fuelCar.getIsPurchased() == true)
                    fuelCar.setCarColor(carColor)
showMessageDialog(frm, "FuelCar has already been purchased : " + distributorName);

                 END IF
                 END IF
                 ELSE
                    fuelCar.setDistributorName(distributorName)
                    fuelCar.setTransmissionType(transmissionType)
                    JOptionPane.showMessageDialog(frm, "FuelCar has  been sucessfully
purchased")
```

```
                END ELSE
                fuelCar.purchasedFuelCar(bookedDate, purchasedDate);
              END IF
            END IF
            ELSE
              showMessageDialog(frm, "PurchasedCarId doesnot found");
            END ELSE
          END IF
         END TRY
       CATCH(Exception ex)
          SHOWDIALOG(, "Invalid Data! please enter a valid Data");
       END CATCH
      END IF
     END DO


IF(e.getSource()==ecardisBtn)
       DO
           IF(list.size()<=0)
             SHOW DIALOG(,"Electric Car is not Added");
          END IF
            ELSE
              SHOW DIALOG("Electric Car detail is displayed")
             IF (Car var5:list)
               IF(var5 instanceof ElectricCar)
                  DISPLAY("Electric car is displayed: ")
                   ElectricCar obj5 IS EQUALS TO (ElectricCar)var5
                    CALLobj5.display()
                   SHOW DIALOG(frm,"Electric Car detail is displayed")
```

```
            END ELSE
          END  FOR
        END IF
        END DO

IF (e.getSource()== buyBtn)
DO
      TRY

            INT carId = CONVERT INTO NUMBER(ecartxtCarID1.getText());
            STRING carName = ecartxtCarName1.getText();
            STRING carBrand = ecartxtcarBrand1.getText();
            STRING carColor = ecartxtcarcolor.getText()

        STRINGcustomerName=txtCustomername.getText()
INTbatteryWarranty=CONVERTINTONUMBER(txtBatteryWarranty.getText())
            STRING range = txtRange.getText()
            INT rechargeTime = CONVERT INTONUMBER(txtRechargetime.getText());
            STRING carPrice = ecartxtCarPrice1.getText()

STRINGpurchaseDate=ecarcmbyearPurchaseDate.getSelectedItem().CONVERT  INTO
STRING() +ecarcmbmthPurchaseDate.getSelectedItem().CONVERT INTO STRING   +
ecarcmbdayPurchaseDate.getSelectedItem().CONVERT INTO STRING

            IF (list.isEmpty())
               showMessageDialog(frm, "Carid list is empty")
            END IF
            ELSE

               FOR (Car car:list)
```

```
                    IF (car.getCarId() == carId)
                        IF (car instanceof ElectricCar)
          ElectricCar electricCar = (ElectricCar electricCar.buyElectricCar(customerName,
batteryWarranty,purchaseDate,range,rechargeTime);
                        END IF
                        ELSE
          SHOW DIALOG(frm," The carid isn't found in electric car");
                        END ELSE
                        END IF
                    ELSE
                      showMessageDialog(, "Electriccar Carid is missing")
                    END ELSE
                  END IF
                END  TRY
                END IF

              CATCH(Exception ex)
                SHOW DIALOG(frm, "You have entered some invalid information");
              END IF
            END DO
if(e.getSource() == sellBtn){
          System.out.println("Ecar sell button clicked");
          try
              int carID = Integer.parseInt(ecartxtCarID1.getText());
              if(list.isEmpty()){

  showMessageDialog(frm, "Car List is Empty");
              ELSE
                FOR(Car car : list){
                    IF(car.getCarId() == carID && car instanceof ElectricCar){
```

```
                    String customerName = txtCustomername.getText();
                    ElectricCar ec = (ElectricCar)car;
                    ec.sellElectricCar(customerName);
                    JOptionPane.showMessageDialog(frm, "Selling Electric Car
Sucessful");
                }ELSE{
                    JOptionPane.showMessageDialog(frm, "The Electric car you
mentioned doesnot exixt");
                END ELSE
              END CATCH
            END IF

        CATCH(Exception ee)
            JOptionPane.showMessageDialog(frm, "Invalid Data! please enter a valid
Data");

        END IF
      END DO

      END CATCH
  END IF
```

6.Error detection and correction

The error seems to be an unauthorized steps performed by a user which enables the programmer to function. Errors in coding may commonly undetected till the code is developed or executed. A few of the issues make it virtually impossible to build or run the program. As either a result, errors should have been corrected prior building and maintaining the program.

The process of determining mistakes in transmitted data and synthesizing the actual mistake information is known called error correction. Error correction implies that the act gives people restored and mistake data. The recognition of faults introduced through interference and other obstacles while communication from the sender to the receiver is known as error detection. The act of recognizing errors and replicating the genuine, mistake data is known as error correcting.

6.1Syntax error

a syntax error is the error in the syntax of a coding and the programing which is developed by the user. This error is caught by a compiler program when user compile their program. The error should be fix and compiled the program with no errors to operate and run the program.

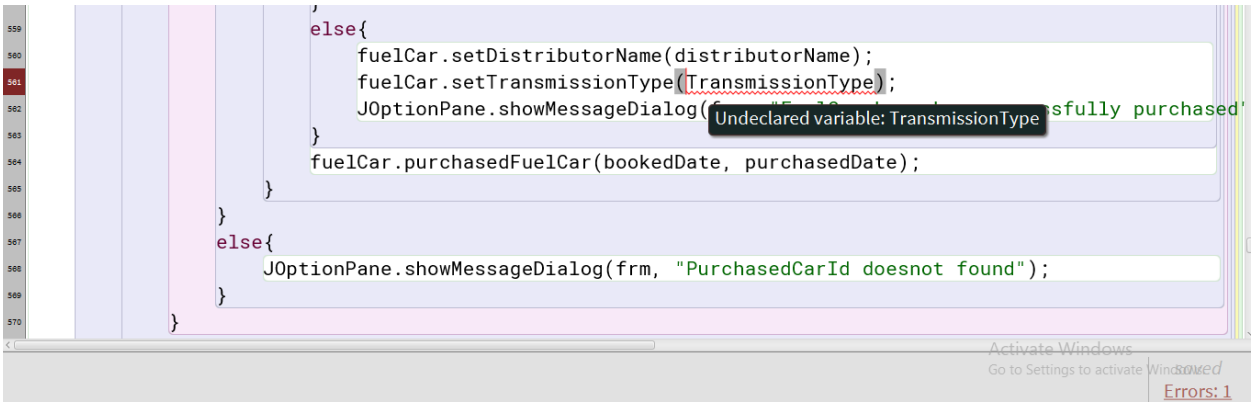


Figure 3Syantax error occur

When I was writing the code of GUI_DEMO class I did not notice that upper case and the lower case of the code are misplaced. The syntax does not match with variable declared

in this class. in above figure we can show the first letter of the word first variable is Transmissiontype shows the error called undeclared varaibles.so in the figure below the undeclared variable Transmissiontype is changed into lowercase Transmissiontype.

```
                }
                else{
                    fuelCar.setDistributorName(distributorName);
                    fuelCar.setTransmissionType(transmissionType);
                    JOptionPane.showMessageDialog(frm, "FuelCar has  been sucessfully purchased
                }
                fuelCar.purchasedFuelCar(bookedDate, purchasedDate);
            }
        }
        else{
            JOptionPane.showMessageDialog(frm, "PurchasedCarId doesnot found");
```

Figure 4 correction of syantax error

6.2Semantic error

Semantic error occurs during the execution of the coding and check either the code is grammatically correct or not. It is occurring in coding while the syntax is correct but putting unusual value is kept in place of some variable. they can be known as undeclared variable. While the modern compiler found the error while compiling the program to execute the code so quickly.

```
        }
    if(e.getSource() == sellBtn){
    System.out.println("Ecar sell button clicked");
    try{
        String carID = Integer.parseInt(ecartxtCarID1.getText());
        if(list.isEmpty()){
            JOptionPane.showMessageDialog(frm, "Car List is Empty");
        }else{
            for(Car car : list){
                if(car.getCarId() == carID && car instanceof ElectricCar){
                    String customerName = txtCustomername.getText();
```

Figure 5Semantic errors occurs

While writing the code in java if we insert datatype of the variable wrong or put undeclared variable then semantic error occur. Hence the datatype of CarId should be String but I accidently put the String datatype. So if we want to run the program smooth we need to change the code to String datatype to int datatype.

```
        }
    if(e.getSource() == sellBtn){
    System.out.println("Ecar sell button clicked");
    try{
        int carID = Integer.parseInt(ecartxtCarID1.getText());
        if(list.isEmpty()){
            JOptionPane.showMessageDialog(frm, "Car List is Empty");
        }else{
            for(Car car : list){
                if(car.getCarId() == carID && car instanceof ElectricCar){
                    String customerName = txtCustomername.getText();
```

Figure 6 correction of the semantic error

6.3Logical error

a logic error is the error in which the code will be complied and it will execute the program also run the whole program but its show wrong output. The java system does not know the output is wrong or right. it is known as the run time error. It is so hard to find because the program will be run in the presence of the error in the code

```
for(Car cars:list){
    if(cars.getCarId() == carId){
        isDuplicatecarId= true;
        break;
    }
}

if(isDuplicatecarId = false){
FuelCar fuel= new FuelCar(carId,carName,carBrand,carPrice,fuelType,numberOfSeats,mileage)
list.add(fuel);
JOptionPane.showMessageDialog(frm,"Car has been added."+ list.size());
}
```

Figure 7 error of logical error

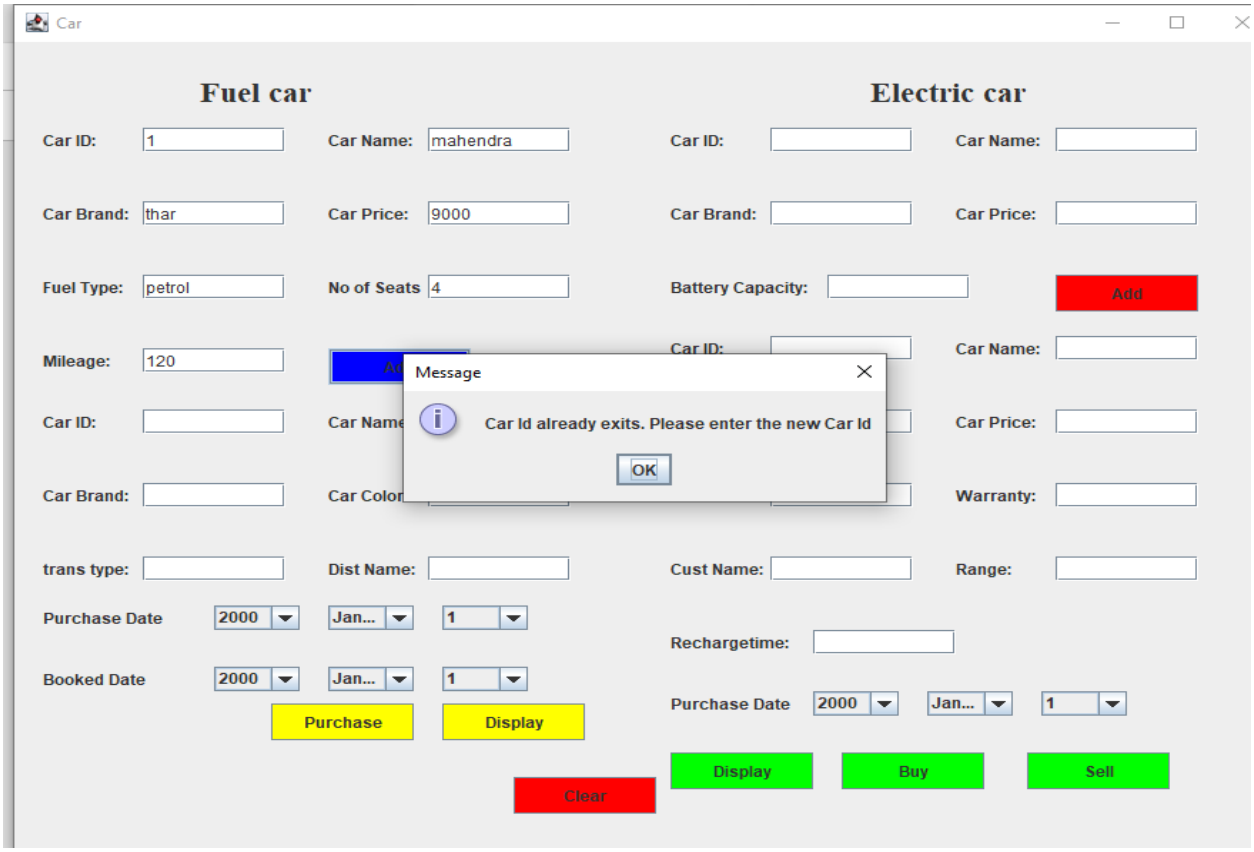
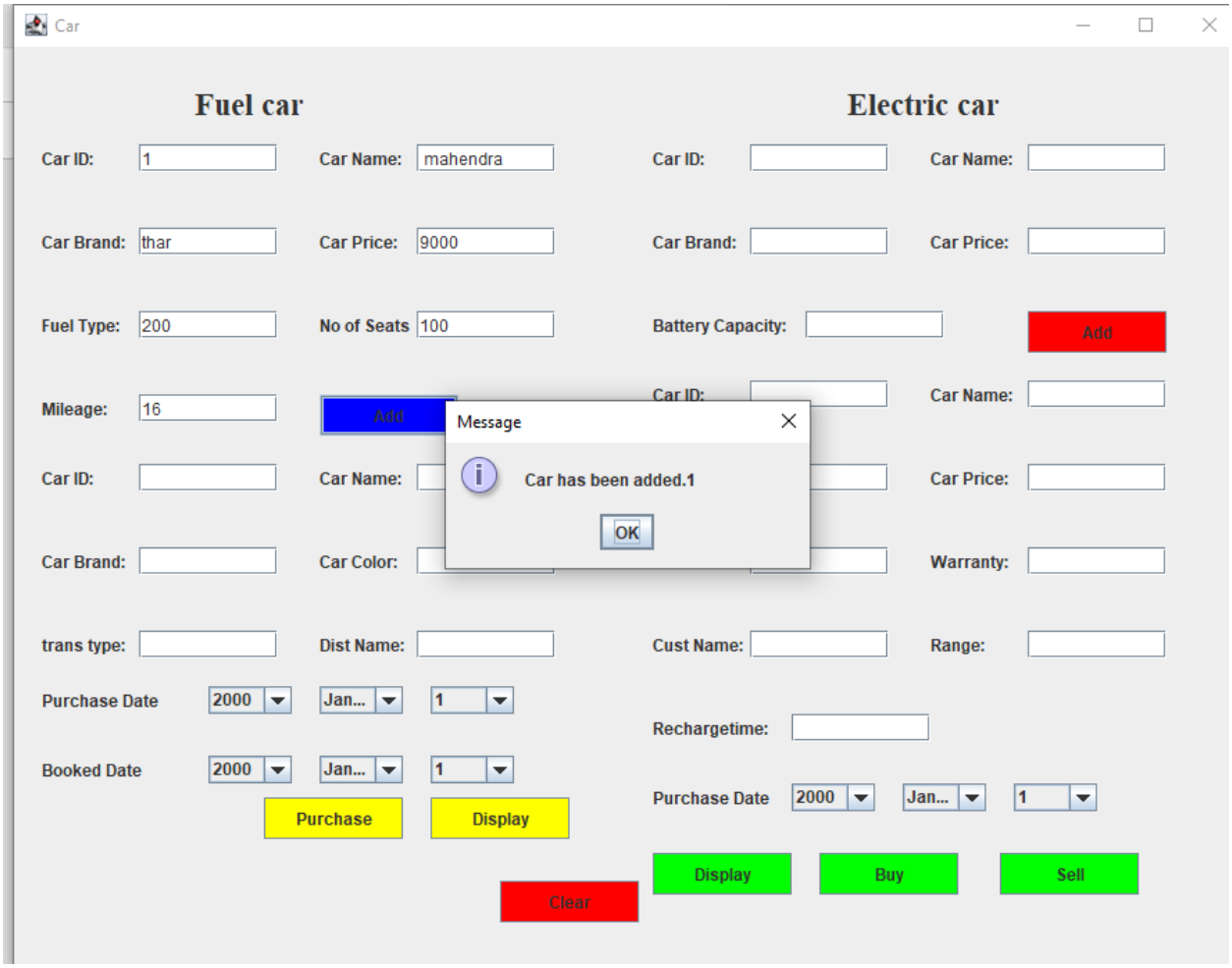


Figure 8 logical errors occurs

In the first figure we can see that (isDuplicatecarId=false). Here a single symbol i.e.”=” assign value to the variable because of which after running the program, the pop up message ”car id already exist .please enter the new car Car id is displayed. This happened because isDuplicateID was assigned false

```
434            }
435
436            if(isDuplicatecarId == false){
437            FuelCar fuel= new FuelCar(carId,carName,carBrand,carPrice,fuelType,numberOfSeats,mileage);
438            list.add(fuel);
439            JOptionPane.showMessageDialog(frm,"Car has been added."+ list.size());
440            }
```

Figure 9 correction of logical error



7.Testing

Test 1: Test that the program can be compiled and run

Table 1 test 1

Test No:	1
Objectives:	Test that the program can be compiled and run
Action:	Open the location of the file in bluej Compile the file Run the file
Expected Result:	The program will be complied and the GUI form should appear
Actual Result:	The program compiled and successfully run
Conclusion:	This test become successful

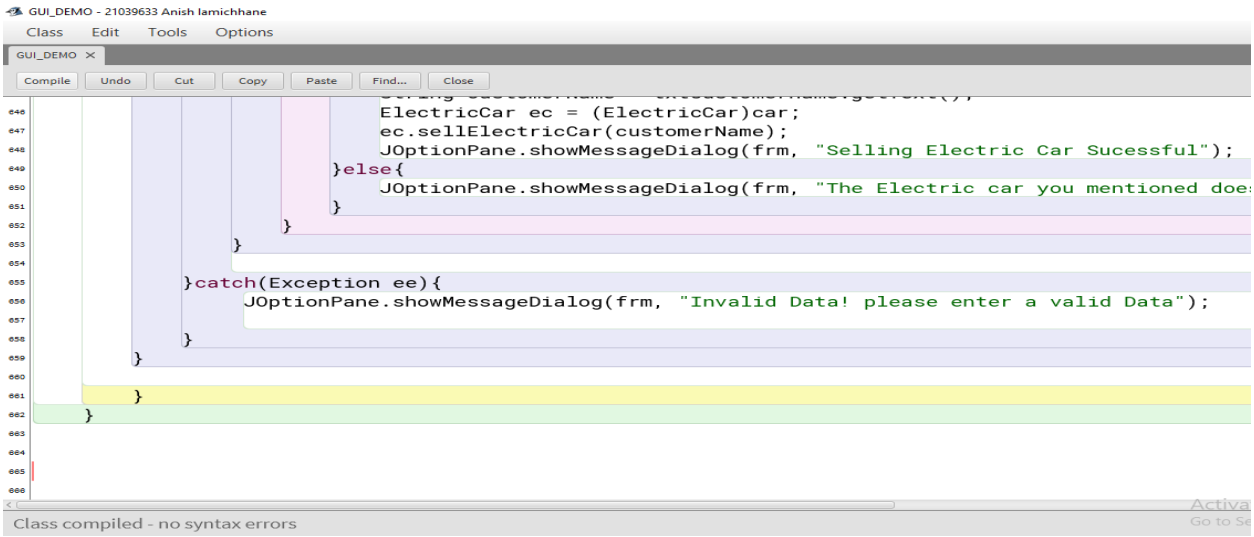


Figure 10 Compiling a program

Figure 11 Run the program

Test 2

Test 2.1 Add Fuel car

Table 2 add fuel car

Test No:	2.1
Objectives:	Add fuel car
Action:	Open the GUI_DEMO FILE of the file in bluej Enter the valid data in carid,carname,carprice,carbrand,fueltype Numbers of seats and mileage And click add button
Expected Result:	Car has been added
Actual Result:	Car has been added
Conclusion:	This test become successful

Figure 12Adding the fcar

Figure 13 after ckicking buy button

Table 3 Add electricCar

Test 2.2 Add electricCar car

Test No:	2.2
Objectives:	Add electricCar car
Action:	Enter the valid data in carid,carname,carprice,carbrand, Batterycapacity Numbers of seats and mileage And click add button
Expected Result:	Car has been added
Actual Result:	Car has been added
Conclusion:	This test become successful

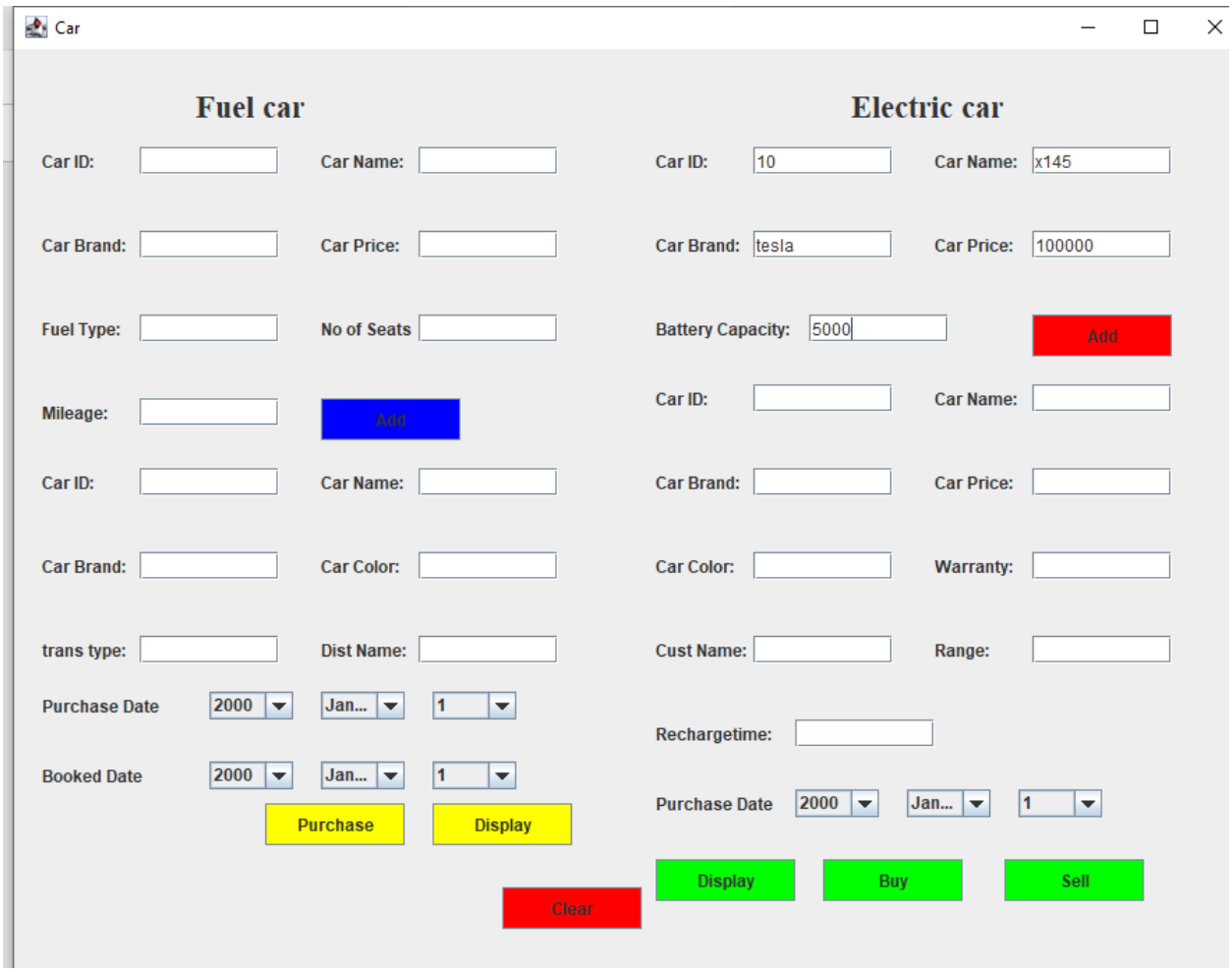


Figure 14 before entering data

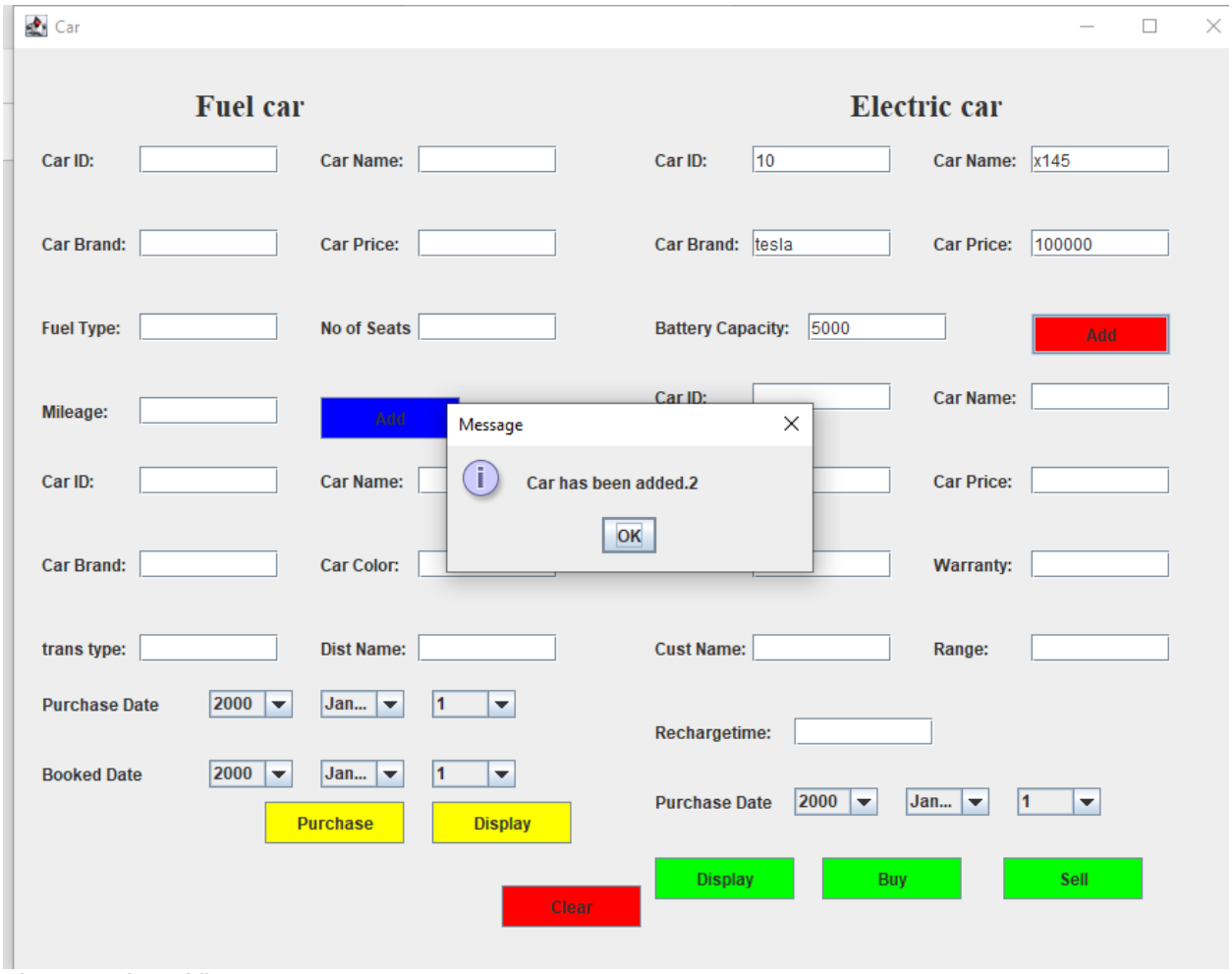


Figure 15 after adding ecar

Test 2.3 Purchase FuelCar

Table 4 purchase fuelcar

Test No:	2.3
Objectives:	Purchase FuelCar
Action:	Add the fuelcar Enter the valid data in carid,carname,carprice,carbrand, purchased date and booked date ,transmission type and distributor name And click purchase button
Expected Result:	Fuelcar is purchased
Actual Result:	FuelCar has been sucessfully purchased
Conclusion:	This test become successful

Figure 16 Purchasing the fcar

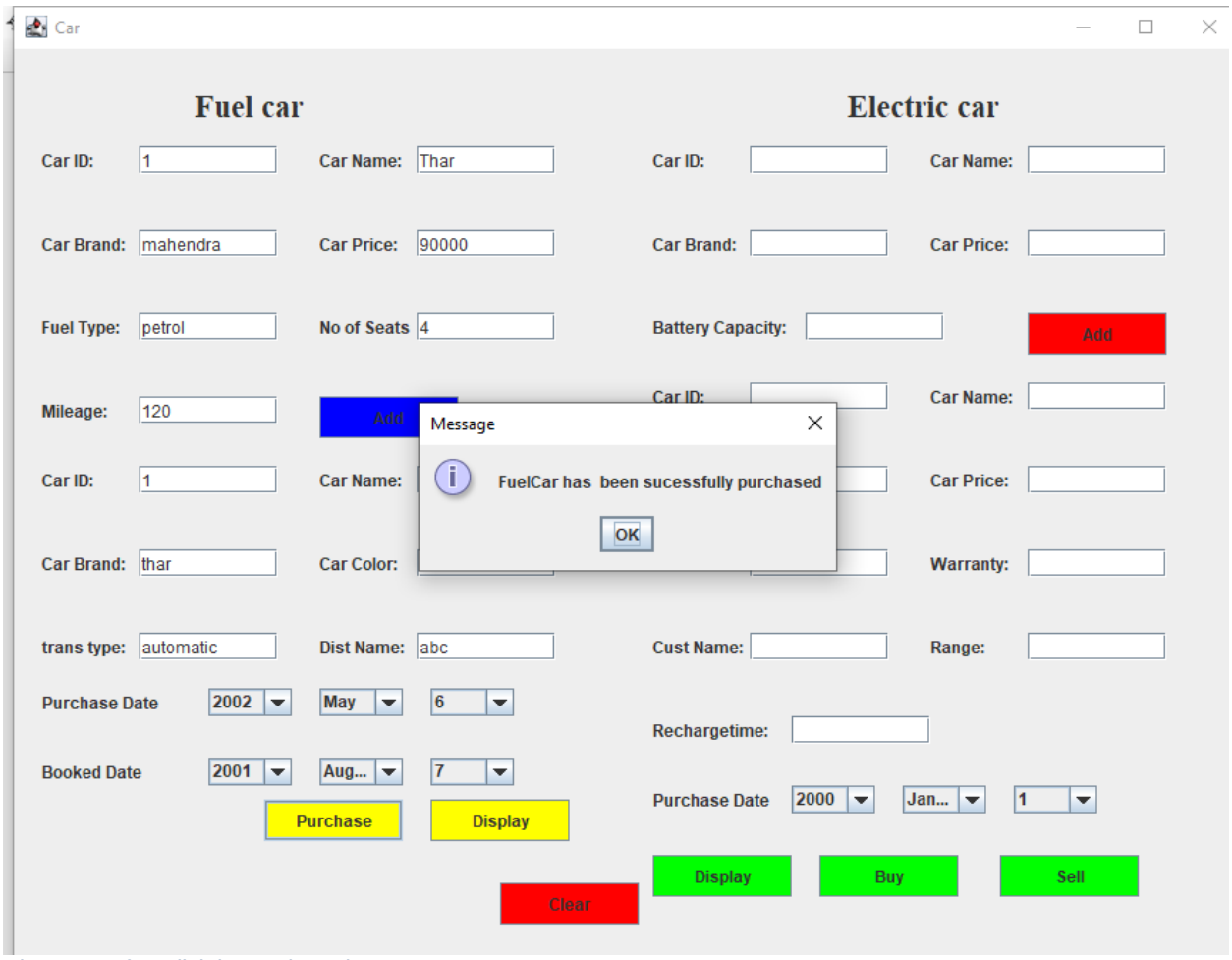


Figure 17 after click in purchase button

Test 2.4 Buy the ElectricCar

Test No:	1
Objectives:	Buy the ElectricCar
Action:	Add the electriccar Enter the valid data in carid,carname,carprice,carbrand, purchased date and customername range ,transmission type and recharge and warranty And click buy button
Expected Result:	Electric car is bought
Actual Result:	Electric car is bought sucessfully purchased
Conclusion:	This test become successful

Table 5 buy the car

Figure 18 filling a buy method


Car
Fuel car
Car ID:
Car Name:
Car Brand:
Car Price:
Fuel Type:
No of Seats
Mileage:
Add
Car ID:
Car Name:
Car Brand:
Car Color:
trans type:
Dist Name:
Purchase Date
2000
Jan...
1
Booked Date
2000
Jan...
1
Purchase
Display
Clear
Electric car
Car ID:
1
Car Name:
tx415
Car Brand:
tesla
Car Price:
10000
Battery Capacity:
1000
Add
Car ID:
1
Car Name:
tx415
Car Brand:
tesla
Car Price:
10000
Car Color:
black
Warranty:
2
Cust Name:
krishna ronaldo
Range:
150
Rechargetime:
2 day
Purchase Date
2020
April
1
Display
Buy
Sell

Figure 19 After clicking buy mehod

2.5 Sell the ElectricCar

Table 6 sell the ecar

Test No:	2.5
Objectives:	Sell the ElectricCar
Action:	Sell the electriccar Enter the valid data in carid, customername And click sell button
Expected Result:	Electric car will sold
Actual Result:	Electric car is sold
Conclusion:	This test become successful

Figure 20 after filling the sell


Car
Fuel car
Car ID:
Car Name:
Car Brand:
Car Price:
Fuel Type:
No of Seats
Mileage:
Add
Car ID:
Car Name:
Car Brand:
Car Color:
trans type:
Dist Name:
Purchase Date
2000
Jan...
1
Booked Date
2000
Jan...
1
Purchase
Display
Clear
Message
The Electric car is sold
OK
Electric car
Car ID:
1
Car Name:
Rangers
Car Brand:
tesla
Car Price:
10000
Battery Capacity:
200
Add
Car ID:
1
Car Name:
Car Price:
Warranty:
Cust Name:
Krishna Ronaldo
Range:
Rechargetime:
Purchase Date
2000
Jan...
1
Display
Buy
Sell

3.1 Trying to add duplicate carID

Test No:	1
Objectives:	Trying to add duplicate carID
Action:	Open the GUI_DEMO FILE of the file in bluej Enter the valid data in carid,carname,carprice,carbrand,fueltype Numbers of seats and mileage and again enter same carid And click add button and again click add button
Expected Result:	Car id is already exist
Actual Result:	Car id is already existing. please enter a new carid.
Conclusion:	This test become successful

Table 7 adding a duplicate carid

Figure 21 filling the fuelcar add button

Figure 22.after clicking add btn of fcar.

Table 8 trying to purchase

3. 2 trying to purchase an already purchased fuelCar.

Test No:	2
Objectives:	trying to purchase an already purchased fuelCar.
Action:	Add the fuelcar Enter the valid data in carid,carname,carprice,carbrand, purchased date and booked date ,transmission type and distributor name And click purchase button and again click the purchase button
Expected Result:	Fuelcar is already purchased
Actual Result:	Fuelcar has been already purchased
Conclusion:	This test become successful

Figure 23 after filling purchase

Figure 24 duplicate of fuelcar

3.3 Trying to sell the ElectricCar which has already been sold

Table 9 try to sell the Electriccar

Test No:	3
Objectives:	Trying to sell the ElectricCar which has already been sold
Action:	Add the electricCar Enter the valid data in carid,carname,carprice,carbrand,battery capacity and customer name and again enter same carid And click the sell button and again click sell button And click buy button and again click thesell button
Expected Result:	electricCar will been sold
Actual Result:	electricCar has been sold
Conclusion:	This test become successful

Figure 25 filling the electric car for duplicate sold

Figure 26 after clicking selling a duplicate id

Test 3.4: . Trying to purchase an already purchased ElectricCar.

Table 10 purcahse an already purrcar

Test No:	4
Objectives:	.Trying to purchase an already purchased ElectricCar.
Action:	Add the electricCar Enter the valid data in carid,carname,carprice,carbrand, purchased date and range recharge time ,warranty,t and customer name And click buy button and again click the buy button
Expected Result:	Fuelcar is already bought
Actual Result:	Fuelcar has been already bought
Conclusion:	This test become successful

Figure 27 after filling buying a duplicate id

Figure 28 display clicking bought a duplicate id

8.Conclution

The report gives an overview of Java and its use to create solutions, and also the capabilities that is used to run the program and design a graphical user interface (GUI) for that. It was a brief summary of the topic.

The report's major aim was to create a graphical user interface (GUI) for collecting the information entered a presenting a satisfactory response The coursework assisted me to gain a stronger insight of the Java programming language. This

Coursework assisted in acquainting myself with Java packages and understanding more about by extended inheritance and implementing interfaces This coursework definitely assisted broaden you GUI knowledge by learning further about JComponents and also how they work. The management of events and exceptions is handled out. While you progress through the courses, you will face many obstacles in this coursework I used the term inheritance. it helps to create acquire the concept of one class to another. It is the process by the which objects of one class acquires the similar properties of objects of another class from which they are derived. In OOP, the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. This is possible by deriving a new class from existing one. Using Java, reducing or downcasting occurs when a subclass property relates to a superclass objects. Downcasting, in other terms, is the process of transforming a subclass type into a superclass type. coursework shows a subclass reference linking to a superclass object.

9.apppendix

```
//importing packages
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
import java.awt.*;

//creating a class to make a GUI form for Fuelcar and electricCar
public class GUI_DEMO implements ActionListener
{
    // declaring attributes of JFrame, JLabel, JTextField, JcomboBox, JButton class
    public JFrame frm;
    public JButton clrBtn;

    //..................fuel car...........
    public JLabel
lblTitlefuel,lblCarIDfuel,lblCarNamefuel,lblCarBrandfuel,lblCarPricefuel,lblCarColorfuel,l
blDistributorname,lblFuelType,lblNumberOfSeats,

lblMileage,lblTransmissionType,lblPurchaseDate,lblBookedDate,lblCarIDfuel1,lblCarBra
ndfuel1,lblCarNamefuel1;
    public JTextField
txtCarIDfuel,txtcarNamefuel,txtcarBrandfuel,txtCarPricefuel,txtCarColorfuel,txtDistributor
name,txtFuelType,txtNumberofSeats,
    txtMileage,txtTransmissionType,txtCarIDfuel1,txtcarNamefuel1,txtcarBrandfuel1;
    public JButton fcaraddBtn,fcarpurBtn,fcardisBtn;
    public JComboBox
cmbyearPurchaseDate,cmbmthPurchaseDate,cmbdayPurchaseDate,cmbyearBookedD
ate,cmbdayBookedDate,cmbmthBookedDate;
    //..................Electric car.................
```

```
    public JLabel
ecarlblTitle,ecarlblCarID,ecarlblcarName,ecarlblcarBrand,ecarlblPurchaseDate,ecarlblC
arPrice,lblCustomername,ecarlblCarPrice1,

lblBatteryCapacity,lblBatteryWarranty,lblRange,ecarlblcarcolor,lblRechargetime,ecarlbl
CarID1,ecarlblcarName1,ecarlblcarBrand1;
    public JTextField
ecartxtCarID,ecartxtCarName,ecartxtcarBrand,ecartxtCarPrice,txtCustomername,txtBatt
eryCapacity,txtBatteryWarranty,

txtRange,txtRechargetime,ecartxtCarID1,ecartxtcarcolor,ecartxtCarName1,ecartxtcarBr
and1,ecartxtCarPrice1;
    public JButton ecaraddBtn,sellBtn,buyBtn,ecardisBtn;
    public JComboBox
ecarcmbyearPurchaseDate,ecarcmbmthPurchaseDate,ecarcmbdayPurchaseDate;

    //Creating Array list of car to store data of GUI form
    //Declration of arraylist
    public ArrayList<Car> list= new ArrayList();
    FuelCar fuel;
    ElectricCar electric;

    // creating main method
    public static void main(String[] args)
    {
        new GUI_DEMO().CarForm();

    }
    // creating a method name CarForm to make GUI form
    public void CarForm()
```

```
{
    //making window frame for Carform
    frm=new JFrame("Car");

    lblTitlefuel =new JLabel();
    lblTitlefuel.setText("Fuel car ");
    lblTitlefuel.setBounds(130, 20, 500, 40);
    frm.add(lblTitlefuel);
    Font ff=new Font("Serif",Font.BOLD,22);
    lblTitlefuel.setFont(ff);

    lblCarIDfuel=new JLabel("Car ID:");
    lblCarIDfuel.setBounds(20,70,100,20);
    frm.add(lblCarIDfuel);

    txtCarIDfuel=new JTextField("");
    txtCarIDfuel.setBounds(90,70,100,20);
    frm.add(txtCarIDfuel);

    lblCarNamefuel=new JLabel("Car Name:");
    lblCarNamefuel.setBounds(220, 70, 100, 20);
    frm.add(lblCarNamefuel);

    txtcarNamefuel=new JTextField("");
    txtcarNamefuel.setBounds(290,70,100,20);
    frm.add(txtcarNamefuel);

    lblCarBrandfuel=new JLabel("Car Brand:");
    lblCarBrandfuel.setBounds(20,130,150,20);
    frm.add(lblCarBrandfuel);
```

```
txtcarBrandfuel=new JTextField("");
txtcarBrandfuel.setBounds(90,130,100,20);
frm.add(txtcarBrandfuel);

lblCarPricefuel=new JLabel("Car Price:");
lblCarPricefuel.setBounds(220,130,150,20);
frm.add(lblCarPricefuel);

txtCarPricefuel=new JTextField("");
txtCarPricefuel.setBounds(290,130,100,20);
frm.add(txtCarPricefuel);

lblFuelType=new JLabel("Fuel Type:");
lblFuelType.setBounds(20,190,200,20);
frm.add(lblFuelType);

txtFuelType=new JTextField("");
txtFuelType.setBounds(90,190,100,20);
frm.add(txtFuelType);

lblNumberOfSeats=new JLabel("No of Seats");
lblNumberOfSeats.setBounds(220,190,200,20);
frm.add(lblNumberOfSeats);

txtNumberofSeats=new JTextField("");
txtNumberofSeats.setBounds(290,190,100,20);
frm.add( txtNumberofSeats);

lblMileage=new JLabel("Mileage:");
```

```
lblMileage.setBounds(20,250,250,20);
frm.add(lblMileage);

txtMileage=new JTextField("");
txtMileage.setBounds(90,250,100,20);
frm.add(txtMileage);

fcaraddBtn = new JButton("Add");
fcaraddBtn.setBackground(Color.BLUE);
fcaraddBtn.setBounds(220,250,100,30);
frm.add(fcaraddBtn);
fcaraddBtn.addActionListener(this);
// for puchase.......................//
lblCarIDfuel1=new JLabel("Car ID:");
lblCarIDfuel1.setBounds(20,300,300,20);
frm.add(lblCarIDfuel1);

txtCarIDfuel1=new JTextField("");
txtCarIDfuel1.setBounds(90,300,100,20);
frm.add(txtCarIDfuel1);

lblCarNamefuel1=new JLabel("Car Name:");
lblCarNamefuel1.setBounds(220, 300, 300, 20);
frm.add(lblCarNamefuel1);

txtcarNamefuel1=new JTextField("");
txtcarNamefuel1.setBounds(290,300,100,20);
frm.add(txtcarNamefuel1);

lblCarBrandfuel1=new JLabel("Car Brand:");
```

```
lblCarBrandfuel1.setBounds(20,360,300,20);
frm.add(lblCarBrandfuel1);

txtcarBrandfuel1=new JTextField("");
txtcarBrandfuel1.setBounds(90,360,100,20);
frm.add(txtcarBrandfuel1);

lblCarColorfuel=new JLabel("Car Color:");
lblCarColorfuel.setBounds(220,360,300,20);
frm.add(lblCarColorfuel);

txtCarColorfuel=new JTextField("");
txtCarColorfuel.setBounds(290,360,100,20);
frm.add(txtCarColorfuel);

lblTransmissionType=new JLabel("trans type:");
lblTransmissionType.setBounds(20,420,100,20);
frm.add(lblTransmissionType);

txtTransmissionType=new JTextField("");
txtTransmissionType.setBounds(90,420,100,20);
frm.add(txtTransmissionType);

lblDistributorname=new JLabel("Dist Name:");
lblDistributorname.setBounds(220,420,100,20);
frm.add(lblDistributorname);

txtDistributorname=new JTextField("");
txtDistributorname.setBounds(290,420,100,20);
frm.add(txtDistributorname);
```

```
lblPurchaseDate=new JLabel("Purchase Date");
lblPurchaseDate.setBounds(20,460,100,20);
frm.add(lblPurchaseDate);

String[] puryear = { "2000", "2001", "2002", "2003", "2004", "2005", "2006","2007","2008", "2009", "2010", "2011","2012",
        "2013", "2014", "2015", "2016", "2017" ,"2018", "2019", "2020", "2021","2022"};
cmbyearPurchaseDate=new JComboBox(puryear);
cmbyearPurchaseDate.setBounds(140,460,60,20);
frm.add(cmbyearPurchaseDate);

String[] purmonth = {"January","February", "March", "April", "May", "June","July", "August", "September",
        "October", "November", "December"};
cmbmthPurchaseDate=new JComboBox(purmonth);
cmbmthPurchaseDate.setBounds(220,460,60,20);
frm.add(cmbmthPurchaseDate);

String[] purday = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14" ,"15", "16", "17", "18", "19",
        "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30","31"};
cmbdayPurchaseDate=new JComboBox(purday);
cmbdayPurchaseDate.setBounds(300,460,60,20);
frm.add(cmbdayPurchaseDate);

lblBookedDate=new JLabel("Booked Date");
lblBookedDate.setBounds(20,510,100,20);
frm.add(lblBookedDate);
```

```
        String[] bookyear = { "2000", "2001", "2002", "2003", "2004", "2005",
"2006","2007","2008", "2009", "2010", "2011","2012",
            "2013", "2014", "2015", "2016", "2017" ,"2018", "2019", "2020", "2021","2022"};
        cmbyearBookedDate=new JComboBox(bookyear);
        cmbyearBookedDate.setBounds(140,510,60,20);
        frm.add(cmbyearBookedDate);

        String[] bookmonth = {"January","February", "March", "April", "May", "June","July",
"August", "September",
            "October", "November", "December"};
        cmbmthBookedDate=new JComboBox(bookmonth);
        cmbmthBookedDate.setBounds(220,510,60,20);
        frm.add(cmbmthBookedDate);

        String[] bookday = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14"
,"15", "16", "17", "18", "19",
            "20", "21", "22", "23", "24", "25","26", "27", "28", "29", "30","31"};
        cmbdayBookedDate=new JComboBox(bookday);
        cmbdayBookedDate.setBounds(300,510,60,20);
        frm.add(cmbdayBookedDate);

        fcarpurBtn = new JButton("Purchase");
        fcarpurBtn.setBackground(Color.yellow);
        fcarpurBtn.setBounds(180,540,100,30);
        frm.add(fcarpurBtn);
        fcarpurBtn.addActionListener(this);

        fcardisBtn = new JButton("Display");
        fcardisBtn.setBackground(Color.yellow);
        fcardisBtn.setBounds(300,540,100,30);
```

```
frm.add(fcardisBtn);
fcardisBtn.addActionListener(this);

//.............................GUI for Electric car.............................
ecarlblTitle =new JLabel();
ecarlblTitle.setText("Electric car ");
ecarlblTitle.setBounds(600, 20, 500, 40);
frm.add(ecarlblTitle);
Font fff=new Font("Serif",Font.BOLD,22);
ecarlblTitle.setFont(fff);

ecarlblCarID=new JLabel("Car ID:");
ecarlblCarID.setBounds(460,70,100,20);
frm.add(ecarlblCarID);

ecartxtCarID=new JTextField("");
ecartxtCarID.setBounds(530,70,100,20);
frm.add(ecartxtCarID);

ecarlblcarName=new JLabel("Car Name:");
ecarlblcarName.setBounds(660, 70, 100, 20);
frm.add(ecarlblcarName);

ecartxtCarName=new JTextField("");
ecartxtCarName.setBounds(730,70,100,20);
frm.add(ecartxtCarName);

ecarlblcarBrand=new JLabel("Car Brand:");
ecarlblcarBrand.setBounds(460,130,150,20);
frm.add(ecarlblcarBrand);
```

```
ecartxtcarBrand=new JTextField("");
ecartxtcarBrand.setBounds(530,130,100,20);
frm.add(ecartxtcarBrand);

ecarlblCarPrice=new JLabel("Car Price:");
ecarlblCarPrice.setBounds(660,130,150,20);
frm.add(ecarlblCarPrice);

ecartxtCarPrice=new JTextField("");
ecartxtCarPrice.setBounds(730,130,100,20);
frm.add(ecartxtCarPrice);

lblBatteryCapacity=new JLabel("Battery Capacity:");
lblBatteryCapacity.setBounds(460,190,150,20);
frm.add(lblBatteryCapacity);

txtBatteryCapacity=new JTextField("");
txtBatteryCapacity.setBounds(570,190,100,20);
frm.add(txtBatteryCapacity);

ecaraddBtn = new JButton("Add");
ecaraddBtn.setBackground(Color.red);
ecaraddBtn.setBounds(730,190,100,30);
frm.add(ecaraddBtn);
ecaraddBtn.addActionListener(this);
// buy and sell for electric car.......................//

ecarlblCarID1=new JLabel("Car ID:");
ecarlblCarID1.setBounds(460,240,100,20);
```

```
frm.add(ecarlblCarID1);

ecartxtCarID1=new JTextField("");
ecartxtCarID1.setBounds(530,240,100,20);
frm.add(ecartxtCarID1);

ecarlblcarName1=new JLabel("Car Name:");
ecarlblcarName1.setBounds(660, 240, 100, 20);
frm.add(ecarlblcarName1);

ecartxtCarName1=new JTextField("");
ecartxtCarName1.setBounds(730,240,100,20);
frm.add(ecartxtCarName1);

ecarlblcarBrand1=new JLabel("Car Brand:");
ecarlblcarBrand1.setBounds(460,300,150,20);
frm.add(ecarlblcarBrand1);

ecartxtcarBrand1=new JTextField("");
ecartxtcarBrand1.setBounds(530,300,100,20);
frm.add(ecartxtcarBrand1);

ecarlblCarPrice1=new JLabel("Car Price:");
ecarlblCarPrice1.setBounds(660,300,150,20);
frm.add(ecarlblCarPrice1);

ecartxtCarPrice1=new JTextField("");
ecartxtCarPrice1.setBounds(730,300,100,20);
frm.add(ecartxtCarPrice1);
```

```
ecarlblcarcolor=new JLabel("Car Color:");
ecarlblcarcolor.setBounds(460,360,150,20);
frm.add(ecarlblcarcolor);

ecartxtcarcolor=new JTextField("");
ecartxtcarcolor.setBounds(530,360,100,20);
frm.add(ecartxtcarcolor);

lblBatteryWarranty=new JLabel("Warranty:");
lblBatteryWarranty.setBounds(660,360,150,20);
frm.add(lblBatteryWarranty);

txtBatteryWarranty=new JTextField("");
txtBatteryWarranty.setBounds(730,360,100,20);
frm.add(txtBatteryWarranty);

lblCustomername=new JLabel("Cust Name:");
lblCustomername.setBounds(460,420,150,20);
frm.add(lblCustomername);

txtCustomername=new JTextField("");
txtCustomername.setBounds(530,420,100,20);
frm.add(txtCustomername);

lblRange=new JLabel("Range:");
lblRange.setBounds(660,420,150,20);
frm.add(lblRange);

txtRange=new JTextField("");
txtRange.setBounds(730,420,100,20);
```

```
frm.add(txtRange);

lblRechargetime=new JLabel("Rechargetime:");
lblRechargetime.setBounds(460,480,150,20);
frm.add(lblRechargetime);

txtRechargetime=new JTextField("");
txtRechargetime.setBounds(560,480,100,20);
frm.add(txtRechargetime);

ecarlblPurchaseDate=new JLabel("Purchase Date");
ecarlblPurchaseDate.setBounds(460,530,100,20);
frm.add(ecarlblPurchaseDate);

String[] purcyear = { "2000", "2001", "2002", "2003", "2004", "2005",
"2006","2007","2008", "2009", "2010", "2011","2012",
        "2013", "2014", "2015", "2016", "2017" ,"2018", "2019", "2020", "2021","2022"};
ecarcmbyearPurchaseDate=new JComboBox(purcyear);
ecarcmbyearPurchaseDate.setBounds(560,530,60,20);
frm.add(ecarcmbyearPurchaseDate);

String[] purcmonth = {"January","February", "March", "April", "May", "June","July",
"August", "September",
        "October", "November", "December"};
ecarcmbmthPurchaseDate=new JComboBox(purcmonth);
ecarcmbmthPurchaseDate.setBounds(640,530,60,20);
frm.add(ecarcmbmthPurchaseDate);

String[] purcday = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14"
,"15", "16", "17", "18", "19",
```

```
        "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30","31"};
    ecarcmbdayPurchaseDate=new JComboBox(purcday);
    ecarcmbdayPurchaseDate.setBounds(720,530,60,20);
    frm.add(ecarcmbdayPurchaseDate);

    buyBtn = new JButton("Buy");
    buyBtn.setBackground(Color.green);
    buyBtn.setBounds(580,580,100,30);
    frm.add(buyBtn);
    buyBtn.addActionListener(this);

    ecardisBtn = new JButton("Display");
    ecardisBtn.setBackground(Color.green);
    ecardisBtn.setBounds(460,580,100,30);
    frm.add(ecardisBtn);
    ecardisBtn.addActionListener(this);

    sellBtn = new JButton("Sell");
    sellBtn.setBackground(Color.green);
    sellBtn.setBounds(710,580,100,30);
    frm.add(sellBtn);
    sellBtn.addActionListener(this);

    clrBtn = new JButton("Clear");
    clrBtn.setBackground(Color.red);
    clrBtn.setBounds(350,600,100,30);
    frm.add(clrBtn);
    clrBtn.addActionListener(this);

    frm.setSize(900,700);
```

```
        frm.setLayout(null);
        frm.setVisible(true);
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == fcaraddBtn){
            int carId=0;
            int numberOfSeats=0;
            int mileage=0;
            String carName="";
            String carBrand="";
            String fuelType="";
            String carPrice="";
            try{
                carId = Integer.parseInt(txtCarIDfuel.getText());
                carName = txtcarNamefuel.getText();
                carBrand = txtcarBrandfuel.getText();
                fuelType = txtFuelType.getText();
                numberOfSeats = Integer.parseInt(txtNumberofSeats.getText());
                mileage = Integer.parseInt(txtMileage.getText());
                carPrice = txtCarPricefuel.getText();
                boolean isDuplicatecarId = false;

                for(Car cars:list){
                    if(cars.getCarId() == carId){
                        isDuplicatecarId= true;
                        break;
```

```
            }
        }

        if(isDuplicatecarId == false){
            FuelCar fuel= new FuelCar(carId,carName,carBrand,carPrice,fuelType,numberOfSeats,mileage);
            list.add(fuel);
            JOptionPane.showMessageDialog(frm,"Car has been added."+ list.size());
        }
        else{
            JOptionPane.showMessageDialog(frm, "Car Id already exits. Please enter the new Car Id");
        }

    }catch(Exception ex){
        JOptionPane.showMessageDialog(frm, "Please check if you have entered valid information");
    }
}
if(e.getSource() == clrBtn)
{
    // clearing textfield of Fuelcar
    txtCarIDfuel.setText("");
    txtcarNamefuel.setText("");
    txtcarBrandfuel.setText("");
    txtCarPricefuel.setText("");
    txtCarColorfuel.setText("");
    txtFuelType.setText("");
    txtNumberofSeats.setText("");
    txtMileage.setText("");
```

```
        txtTransmissionType.setText("");
        txtCarIDfuel1.setText("");
        txtcarNamefuel1.setText("");
        txtcarBrandfuel1.setText("");
        txtDistributorname.setText("");
        // clearing textfield of Electriccar
        ecartxtCarID.setText("");
        ecartxtCarName.setText("");
        ecartxtcarBrand.setText("");
        ecartxtCarPrice.setText("");
        txtBatteryCapacity.setText("");
        txtBatteryWarranty.setText("");
        txtCustomername.setText("");
        txtRange.setText("");
        ecartxtcarcolor.setText("");
        ecartxtCarID1.setText("");
        ecartxtCarName1.setText("");
        ecartxtcarBrand1.setText("");
        ecartxtCarPrice1.setText("");
        txtRechargetime.setText("");

    }
    if(e.getSource()== ecaraddBtn){
        int  carID3 = 0;
        String carName = "";
        String carBrand = "";
        int batteryCapacity = 0;
        String carPrice = "";

        try{
```

```
            int carID= Integer.parseInt(ecartxtCarID.getText());
            String CarName = ecartxtCarName.getText();
            String CarBrand = ecartxtCarID.getText();
            String CarPrice = ecartxtCarID.getText();
            int BatteryCapacity= Integer.parseInt(txtBatteryCapacity.getText());

            boolean carIDFound = false;
            for(Car cars :list){
                if(cars.getCarId()== carID3){
                    carIDFound = true;
                    break;
                }
            }
            if(carIDFound==false){
                ElectricCar electric= new
ElectricCar(carID,CarName,CarBrand,CarPrice,BatteryCapacity);
                list.add(electric);
                JOptionPane.showMessageDialog(frm,"Car has been added."+list.size());

            }else{
                JOptionPane.showMessageDialog(frm,"CarID already exists please enter
new Car ID");
            }

        }catch(Exception ex1){
            JOptionPane.showMessageDialog(frm,"Please Enter Valid Informations");
        }
    }

    if(e.getSource()==fcardisBtn)
```

```
        {
            if(list.size()<=0){
                JOptionPane.showMessageDialog(frm,"FuelCar is not Added");
            }else{
                JOptionPane.showMessageDialog(frm,"Fuel Car detail is displayed");

                for(Car var5:list){
                    if(var5 instanceof FuelCar){
                        System.out.println("fuelcar is displayed");
                        FuelCar fuel=(FuelCar) var5;
                        fuel.display();

                    }
                }
            }
        }
        if(e.getSource() == fcarpurBtn){

            try{
                int carId = Integer.parseInt(txtCarIDfuel1.getText());
                String carName = txtcarNamefuel1.getText();
                String carBrand = txtcarBrandfuel1.getText();
                String carColor = txtCarColorfuel.getText();
                String transmissionType = txtTransmissionType.getText();
                String distributorName = txtDistributorname.getText();
                String purchasedDate =
String.valueOf(cmbyearPurchaseDate.getSelectedItem())+String.valueOf(cmbmthPurch
aseDate.getSelectedItem()) + String.valueOf(cmbdayPurchaseDate.getSelectedItem());
```

```
            String bookedDate =
String.valueOf(cmbyearBookedDate.getSelectedItem())+String.valueOf(cmbmthBooked
Date.getSelectedItem()) + String.valueOf(cmbdayBookedDate.getSelectedItem());
            boolean isCarIdThere = false;
            for(Car car:list){
                if(car.getCarId() == carId){
                    if(car instanceof FuelCar){
                        FuelCar fuelCar = (FuelCar)car;
                        if(fuelCar.getIsPurchased() == true){
                            fuelCar.setCarColor(carColor);
                            JOptionPane.showMessageDialog(frm, "FuelCar has already been
purchased : " + distributorName);

                        }
                        else{
                            fuelCar.setDistributorName(distributorName);
                            fuelCar.setTransmissionType(transmissionType);
                            JOptionPane.showMessageDialog(frm, "FuelCar has been
sucessfully purchased");
                        }
                        fuelCar.purchasedFuelCar(bookedDate, purchasedDate);
                    }
                }
                else{
                    JOptionPane.showMessageDialog(frm, "PurchasedCarId doesnot
found");
                }
            }
        }catch(Exception ex){
```

```
            JOptionPane.showMessageDialog(frm, "Invalid Data! please enter a valid Data");
        }
    }
    //event handling for the Electric car//
    if(e.getSource()==ecardisBtn)
    {
        if(list.size()<=0){
            JOptionPane.showMessageDialog(frm,"Electric Car is not Added");
        }else{
            JOptionPane.showMessageDialog(frm,"Electric Car detail is displayed");
            for(Car var5:list){
                if(var5 instanceof ElectricCar){
                    System.out.println("Electric car is displayed: ");
                    ElectricCar obj5=(ElectricCar)var5;
                    obj5.display();
                    JOptionPane.showMessageDialog(frm,"Electric Car detail is displayed");

                }
            }
        }
    }

    if(e.getSource()== buyBtn){
        System.out.println("Buy button is pressed");
        try{
            int carID = Integer.parseInt(ecartxtCarID1.getText());
            if(list.isEmpty()){
                JOptionPane.showMessageDialog(frm,"Car List is Empty");
            }else{
```

```
            boolean isThere =false;
            for(Car car:list){
                System.out.println(car.getCarName());
                if(car.getCarId()==carID && car instanceof ElectricCar){

                    ElectricCar electricCar =(ElectricCar)car;
                    String boughtDate =
String.valueOf(ecarcmbyearPurchaseDate.getSelectedItem())+String.valueOf(ecarcmb
mthPurchaseDate.getSelectedItem()) +
String.valueOf(ecarcmbdayPurchaseDate.getSelectedItem());
                    String customerName = txtCustomername.getText();
                    int Warranty = Integer.parseInt(txtBatteryWarranty.getText());
                    String range = txtRange.getText();
                    int rechargeTime = Integer.parseInt(txtRechargetime.getText());
                    if(electricCar.getIsBought() == false){
                        JOptionPane.showMessageDialog(frm,"Electric Car has been
Bought");

                    } else{
                        JOptionPane.showMessageDialog(frm,"Electric Car has been
already Bought");
                    }
                    electricCar.buyElectricCar(customerName, Warranty, boughtDate,
range, rechargeTime);
                    isThere=false;
                    break;
                }

                else{
                    isThere = true;
```

```
                }

            }
                if(isThere == true){
                    JOptionPane.showMessageDialog(frm, "No Record Found ");
                }

            }

        }
        catch(Exception ex2){
            JOptionPane.showMessageDialog(frm,"Please Enter the Valid ID. ID should be
Number");

        }
    }


    if(e.getSource()== sellBtn){
        System.out.println("Sell button is pressed");
        try{
            int carID = Integer.parseInt(ecartxtCarID1.getText());
            if(list.isEmpty()){
                JOptionPane.showMessageDialog(frm,"Car List is Empty");
            }else{
                boolean isThere = false;
                for(Car car:list){
                    if(car.getCarId() == carID && car instanceof ElectricCar){
```

```
                    ElectricCar electricCar =(ElectricCar)car;

                     if(electricCar.getIsSold() == false){
                         electricCar.sellElectricCar( txtCustomername.getText());
                        JOptionPane.showMessageDialog(frm,"Electric   Car   has   been
sold");

                    } else{
                       JOptionPane.showMessageDialog(frm,"Electric   Car   has   been
already sold");
                    }
                    isThere=false;
                    break;


                }
                else
                   isThere = true;

            }
            if(isThere == true){
                JOptionPane.showMessageDialog(frm, "No Record Found ");
            }

          }

        }
```

```
            catch(Exception ex3){
                JOptionPane.showMessageDialog(frm,"Please Enter the Valid ID. ID should be
Number");

            }
        }
    }
}



/**
 * Write a description of class Car here.
 *
 * @author (21039633 Anish lamichhane)
 * @version (1.0.0)
 */
// declaring 5 attrubutes of Car using (private) access modifier to use within this class only
public class Car
{
    private int carId;
    private String carName;
    private String carBrand;
    private String carPrice;
    private String carColor;
    // declearing constructor which contain parameter like carID, carName, carBrand,
carPrice, carColor
    public Car(int carId, String carName, String carBrand, String carPrice){
        //attributes = parameter
    //to set the value for instance variables this. is used
```

```
        this.carName = carName;
        this.carId = carId;
        this.carBrand = carBrand;
        this.carPrice = carPrice;
        this.carColor = "";
    }

    // creating accessor method name carId which return initialized value of instance variable
    public int getCarId(){
        return this.carId;
    }
        // creating accessor method name carName which return initialized value of instance variable
    public String getCarName(){
        return this.carName;
    }
    // creating accessor method name carBrand which return initialized value of instance variable
    public String getCarBrand(){
        return this.carBrand;
    }
        // creating accessor method name carColor which return initialized value of instance variable
    public String getCarPrice(){
        return this.carPrice;
    }

    public String getCarColor(){
        return this.carColor;
```

```
    }

        // creating mutator method name carColor to set the color of the car which it's default
is null and not initialized in
    //constructor

    public void setCarColor(String carColor){
        this.carColor = carColor;
    }
        //(a method name display is created in this class the attributes
      //carId,carName, carBrand, carPrice, carColor is displayed with proper data entry)



    public void display(){
        System.out.println("Car ID:  "+carId);
        System.out.println("Car Name:  "+carName);
        System.out.println("Car Brand:  "+carBrand);
        System.out.println("Car Price:  "+carPrice);
        if (carColor==""){
            System.out.println("the car color is empty");

          }
          else{
            System.out.println("the car color is "+carColor);
          }
    }
}

/**
 * Write a description of class Car here.
```

```
 *
 * @author (21039633 Anish lamichhane)
 * @version (1.0.0)
 */

public class ElectricCar extends Car{
    //attributes of class ElectricCar
    private String customerName;
    private int batteryCapacity;
    private int batteryWarranty;
    private String purchaseDate;
    private String range;
    private int rechargeTime;
    private boolean isBought;
    private boolean isSold;

    //Creation of contrustor for class ElectricCar
    public ElectricCar(int carId,String carName,String carBrand,String carPrice,int
batteryCapacity){
        //constructor of parent class inherited must be initialized from child class which is
done using super keyword
        super(carId, carName, carBrand, carPrice);
        this.batteryCapacity = batteryCapacity;

        this.customerName = "";
        this.batteryWarranty = 0;
        this.purchaseDate = "";
        this.range = "";
        this.rechargeTime = 0;
        this.isBought = false;
```

```
        this.isSold = false;
    }
        //getter of the data of the class ElectricCar
    public String getCustomerName(){
        return this.customerName;
    }

    public int getBatteryCapcity(){
        return this.batteryCapacity;
    }

    public int getBatteryWarrenty(){
        return this.batteryWarranty;
    }

    public String getPurchaseDate(){
        return this.purchaseDate;
    }

    public String getRange(){
        return this.range;
    }

    public int getRechargeTime(){
        return this.rechargeTime;
    }

    public boolean getIsBought(){
        return this.isBought;
    }
```

```
    public boolean getIsSold(){
        return this.isSold;
    }

    //setter/Mutator of the data of class ElectricCar
    public void setCustomerName(String customerName){
            // setting customer name only if car is not bought.
        if(isBought == false){
            this.customerName = customerName;
        }else{
            System.out.println("The Electric car is already bought so customer name cannot be assigned");
        }
        }
        // Creating a method to buyElectricCar
    public void buyElectricCar(String customerName, int batteryWarranty, String purchaseDate, String range, int rechargeTime){
            if(isBought == false){
            setCustomerName(customerName);
            this.batteryWarranty = batteryWarranty;
            this.purchaseDate = purchaseDate;
            this.range = range;
            this.rechargeTime = rechargeTime;
            this.isBought = true;
        }else{
            System.out.println("The Electric car has already been bought");
        }
        }
        //Creating a method to sellElectricCar
```

```
public void sellElectricCar(String customerName){
        if(isSold == false){
        // setting new customer name after electric car sold
        this.customerName = customerName;
        this.batteryCapacity = 0;
        this.batteryWarranty = 0;
        this.purchaseDate = "";
        this.range = "";
        this.rechargeTime = 0;
        this.isSold = true;
        this.isBought = false;
    }else{
        System.out.println("The Electric car has already been sold");
    }

    }
    //Creating display method for class ElectricCar
@Override
public void display(){
        super.display();
        // if(isBought == true){
        System.out.println("Customer Name: "+customerName);
        System.out.println("Battery Capacity: "+batteryCapacity);
        System.out.println("Battery Warranty: "+batteryWarranty);
        System.out.println("Purchased Date: "+purchaseDate);
        System.out.println("Range: "+range);
        System.out.println("Recharge Time: "+rechargeTime);
    }
}
```

```
/**
 * Write a description of class FuelCar here.
 *
 * @author (21039633 Anish lamichhane)
 * @version (1.0.0)
 */
 //creating a sub or child class name fuelcar of super class Car
public class FuelCar extends Car
{
     //state 8 attributes using (private) access modifier to use in FuelCar class only
    private String distributorName;
    private String fuelType;
    private int numberOfSeats;
    private String bookedDate;
    private String purchasedDate;
    private int mileage;
    private String transmissionType;
    private boolean isPurchased;
    //creating constructor having parameter like
distributorName,fuelType,numberofSeats,bookedDate
     //purchasedDate,mileage,transmissionType,isPurchased

    public FuelCar(int carId, String carName, String carBrand, String carPrice, String
fuelType,
    int numberOfSeats, int mileage){
    //for accessing instance variable and methods from super class (super) keyword is
used
       super(carId,carName,carBrand,carPrice);
        this.fuelType = fuelType;
```

```
        this.numberOfSeats = numberOfSeats;
        this.mileage = mileage;
     //set the empty string to the rest of the string variable and false to the boolean
        this.distributorName = "";
        this.bookedDate = "";
        this.purchasedDate = "";
        this.transmissionType = "";
        this.isPurchased = false;
    }
    // creating accessor method name distributorName which return initialized value of instance variable
    public String getDistributorName(){
        return this.distributorName;
    }
     //creating accessor method name FuelType which return initialized value of instance variable
    public String getFuelType(){
        return this.fuelType;
    }
    //creating accessor method name getNumberOfSeats which return initialized value of instance variable
    public int getNumberOfSeats(){
        return this.numberOfSeats;
    }
    //creating accessor method name BookedDate which return initialized value of instance variable
    public String getBookedDate(){
        return this.bookedDate;
    }
```

```
    //creating accessor method name PurchasedDate which return initialized value of instance variable
    public String getPurchasedDate(){
        return this.purchasedDate;
    }
    //creating accessor method name Mileage which return initialized value of instance variable
    public int getMileage(){
        return this.mileage;
    }
    //creating accessor method name TransmissionType which return initialized value of instance variable
    public String getTransmissionType(){
        return this.transmissionType;
    }
    //creating accessor method name IsPurchased which return initialized value of instance variable
    public boolean getIsPurchased(){
        return this.isPurchased;
    }

     //creating mutator method name TransmissionType to set value of type of Transmission
    public void setTransmissionType(String transmissionType){
        this.transmissionType = transmissionType;
    }
     //creating mutator method name DistributorName to set name of Distributor
    public void setDistributorName(String distributorName){
        this.distributorName = distributorName;
    }
```

```
 //creating mutator method name IsPurchased to set value of IsPurchased
public void setIsPurchased(boolean isPurchased){
    this.isPurchased = isPurchased;
}
    /*creating mutator method name String bookedDate, String purchasedDate
* check if joined is true a messege is display
* else value of String bookedDate, String purchasedDate is set
*/

public void purchasedFuelCar(String bookedDate, String purchasedDate){
    if( isPurchased == true){
        setCarColor(getCarColor());
        super.display();
          System.out.println("id:" + getCarId());
        System.out.println("name:"+ getCarName());
        System.out.println("brand:"+ getCarBrand());
        System.out.println("price:"+ getCarPrice());
        System.out.println("distributorName "+distributorName);
    }else{
        setDistributorName(distributorName);
        setTransmissionType(transmissionType);
        this.bookedDate = bookedDate;
        this.purchasedDate = purchasedDate;
        isPurchased = true;
    }
}
//method name display is created
//display method of super class is called using super key word
//value of
distributorName,fuelType,numberofSeats,bookedDate,purchasedDate,mileage,
```

```
 //transmissionType,isPurchasedis display with proper messege
public void display(){
 if(isPurchased == true){
        super.display();
        System.out.println("Distributor Name: "+distributorName);
        System.out.println("Fuel Type: "+fuelType);
        System.out.println("Booked Date: "+bookedDate);
        System.out.println("Purchased Date: "+purchasedDate);
        System.out.println("Mileage: "+mileage);
        System.out.println("Number of Seats: "+numberOfSeats);
        System.out.println("Transmission Type: "+transmissionType);
    }

        //}

}
}
```