**IT 114 - ADVANCED PROGRAMMING FOR INFORMATION TECHNOLOGY**

**Case Studies**

**1. Formulating the Problem**

**1.1 Problem Description**

We must create a program that creates a GUI that allows the user to design/build your own car. It displays the various amenities, colors, brands, and models, you can choose from and outputs the price added up.

The user should have the ability to choose from a various selection of car brands, models, colors, and amenities. The program should tally up all these variables and display the costs and specs to the customer, along with the customer name in the display

**1.2 Verbalization**

*What is the goal?*

Create a program that allows you to design/build your own car. It displays the various amenities, colors, brands, and models, you can choose from and outputs the price added up

*What are the givens?*   The types of cars, models, colors, amenities, and prices of each amenity, car and model

*What are the unknowns?*   The total price of the build, customer name

1.3 **Information Elicitation**

*Goal*

Create a program that – allows you to design a car --- allows the user to choose a brand --- allows user to choose a model --- allows user to choose color --- allows user to choose additional amenities.

*Givens*   The car brands, the car models, the car colors, the car amenities, and the pricing for each of those

*Unknowns*   Who is the customer, what will be the total price of the build

*Conditions*   The total price must be correct.

**2. Planning the Solution**

**2.1 Solution Strategy**

Create a Combo Box that contains the Car brand. Create a Combo Box that contains the Car model. Create a Combo box that contains the Car colors the customer can choose from. Create labels for each Combo Box, Category, Buttons, and Check Boxes (Customer Name, Car Brand, Car Model, Car Color, Amenities/Add-Ons, Air Conditioned Seats, Automatic Emergency Braking, Backup Camera, Blue-Tooth Connectivity, Heated Seats, Keyless Entry, Navigation System, Super Sound System, Clear, Close Program, Calculate)

Create a Grid layout to hold the check boxes, combo boxes, labels, and text fields/areas. Create the stage and scene. Set the Grid as the layout parameter for the scene. Use Insets to add padding to the grid. Create buttons to either close program, clear the program, or calculate the selected variables.

**)**

**2.2 Goal Decomposition**

*Sub-goal 1*

Create Stage

*Sub-goal 2*

Create Scene.

*Sub-goal 3*

Create Grid Layout.

*Sub-goal 4*

Create a Combo Box that contains the Car brand.

*Sub-goal 5*

Create a Combo Box that contains the Car model.

*Sub-goal 6*

Create a Combo box that contains the Car colors the customer can choose from.

*Sub-goal 7*

Create Close Program Button

*Sub-goal 8*

Create Clear Program Button

*Sub-goal 9*

Create Calculate Button that sums up all the variables and displays the output

*Sub-goal 10*

Create checkboxes for each of the Amenities the Customer can choose form

*Sub-goal 11*

Create labels for each Combo Box, Category, Buttons, and Check Boxes

*Sub-goal 12*

Put all the Node items into the Grid

*Sub-goal 13*

Set the grid as the Layout of the scene

*Sub-goal 13*

Show the stage

**2.3 Resources**

*Relevant Information*

**Car Brand:** Cadillac $30,000, Ford $13,000, GM $15,000, Honda $20,000, Mercedes $45,000, Nissan $14,000

**Model:** Convertible $14,250, Coupe $12,000, Hatchback $10,000, Sedan $15,550, SUV $25,750

**Colors:** Black, Blue, Red, Silver, White

**Amenities:** Heated Seats $750, Air Conditioned Seats $750, Backup Camera $550, Super Sound System $1,200, Navigation System $1100, Blue-Tooth Connectivity $550, Keyless Entry $650, Automatic Emergency Braking $5000

**3. Designing the Solution**

**)3.1 Structure Chart**

*First Level Decomposition*



*Goal Refinement*

**Sub-goal 1**

Create a stage

**Sub-goal 2**

Create a scene

**Sub-goal 3**

Create a Grid Layout

**Sub-goal 4**

Create a ComboBox for Brand, Model, and Color

* The combo boxes will contain the various brands, models, and colors for the user to select from.

**Sub-goal 5**

Create CheckBoxes for all of the Amenities for the user to choose from

* The Checkboxes will contain; Heated Seats $750, Air Conditioned Seats $750, Backup Camera $550, Super Sound System $1,200, Navigation System $1100, Blue-Tooth Connectivity $550, Keyless Entry $650, Automatic Emergency Braking $5000

**Sub-goal 6**

Create a TextField for the user to input their name

* User input will be displayed in the output when the calculation is complete

**Sub-goal 7**

Create a TextArea for the output to be displayed

**Sub-goal 8**

Create labels for the Categories; Customer Name, and Car Brand, Car Model, Amenities, Car Specs and Price(Output)

**Sub-goal 9**

Calculate all of the variables and assign it to variable cost, and put the chosen variables in a string that shows what they got called Output.

**Sub-goal 10**

Add cost to string output for the final Display

**Sub-goal 11**

Add all the nodes to their own areas in the Grid

**Sub-goal 12**

Set the Grid as the layout for the scene

**Sub-goal 13**

Show the Stage

*Second Level Decomposition*



The second level decomposition shows operations between the User and the System. The process starts with the user having an area to insert their name and the choice between; car brand, model, color, and amenities they would like in there. Calculate takes all of the variables that the user input and sums them up. It then puts the variable information and the customer name in an Output string and displays it in the TextArea with the total price. After the user has to option to clear and start again or close the program. The Clear button clears all the input fields that the user has changed back to null. The close program button closes the program and exits the stage.

**3.2 Module and Data Specifications**

**Name**: Choice – Display a TextField “Customer Name” for the user to enter their name

**Input**: 1

**Output**: Customer Name will be printed with the Output at the end

**Logic**: Retrieve the Customer’s name from the TextField

**Name**: Choice - Display ComboBox labeled Car Brand to User to choose a car brand from

**Input**: Cadillac $30,000, Ford $13,000, GM $15,000, Honda $20,000, Mercedes $45,000, Nissan $14,000

**Output**: Cost of car and the name will be added to their respected strings to be totaled at the end

**Logic**: User chooses from a list

**Name**: Choice - Display ComboBox labeled Car Model to User to choose a car model from

**Input**: Convertible $14,250, Coupe $12,000, Hatchback $10,000, Sedan $15,550, SUV $25,750

**Output**: Cost of Car Model and the name will be added to their respected strings to be totaled at the end

**Name**: Choice - Display ComboBox labeled Car Color to User to choose a car color from

**Input**: Black, Blue, Red, Silver, White

**Output**: The car color will be added to the string when displaying the specs of the car at the end

**Name**: Choice - Display CheckBoxes labeled in Category Amenities/Add-Ons for User to choose from

**Input**: Heated Seats $750, Air Conditioned Seats $750, Backup Camera $550, Super Sound System $1,200, Navigation System $1100, Blue-Tooth Connectivity $550, Keyless Entry $650, Automatic Emergency Braking $5000.

**Output**: Cost of car and the name will be added to their respected strings to be totaled at the end

**Name**: Choice – User has option to choose Close Program Button

**Input**: Close Program clicked

**Output**: Program closes

**Logic**: the operation ends and the stage exits.

**Name**: Choice – User has option to choose Clear Button

**Input**: Clear clicked

**Output**: All Input fields and variables changed by user are cleared

**Logic**: All of the inputs fields that the user had changed are set to null or their respective blank variables.

**Name**: Choice – User has option to choose Calculate Button

**Input**: Calculate clicked

**Output**: Total costs along with the car specs

**Logic**: Program takes all the variables the user has selected and adds the costs. It then displays the costs along with the car specs and customer name in the TextArea Field.

**3.3 Logic**

*Logic*

1. Open Stage
2. User Inputs name into TextField

3.0 Select from Variables

4.0 Upon Calculate, sum up all the variables chosen by the user.

4.1 Add all the costs in a String “cost”

4.2 Create a String “Output” that displays the car specs, Customer Name, and total cost.

3.3.1 Add the “cost” String to the String “Output”

3.3.2 Add the Customer Name to the String “Output”

4.3 Display the String “Output” in the TextArea

5.0 Upon Selecting Clear

5.1 Clear all of the variables changed by the user to their default

null/blank values.

6.0 Upon Selecting Close Program

6.1 Exit/Close Program

*Algorithm Description*

The process of doing this : Data is stored in checkboxes and comboboxes. The system gives the User an option to Close Program, Clear or Calculate this is done by pressing their assigned buttons. If the User selects Close Program the Program closes and the stage closes.

Clear, Clears all of the variables, that were changed by the user, to their default null/blank values. After the User has the option to resubmit their selected variables.

Calculate the program the price of all the variables chosen by the customer. It also takes the name the Customer input and puts that in the string “output” along with all of the chosen car variables. The cost is summed up in a string “cost”. The string “cost” is then inserted into string ”output”, where it is then displayed in the TextArea. The user then has the option to Clear or Close program.

**4. Translation**

**4.1** **Source Code**

I removed the indents to try to make it easier to read in this document. With the indents it was hard to look at.

//===================================================   
// Name : Emad Tirmizi  
// SID : 31400222  
// Course : IT114   
// Section :   
// Instructor : Maura Deek  
// T.A :   
//===================================================   
//===================================================   
// Assignment # : 2  
// Date : 10/18/2018  
//===================================================   
//===================================================   
// Description: This program will create create a   
// program that creates a GUI that allows the user to   
// design/build your own car. It displays the various   
// amenities, colors, brands, and models, you can   
// choose from and outputs the price added up.   
// The user should have the ability to choose from a  
// various selection of car brands, models, colors,   
// and amenities. The program should tally up all  
// these variables and display the costs and specs to   
//the customer, along with the customer name in the   
//display  
//===================================================   
  
**import** javafx.scene.layout.GridPane;  
**import** javafx.application.Application;  
**import** javafx.event.ActionEvent;  
**import** javafx.event.EventHandler;  
**import** javafx.stage.Stage;  
**import** javafx.scene.Scene;  
**import** javafx.geometry.Insets;  
**import** javafx.scene.layout.VBox;  
**import** javafx.scene.layout.BorderPane;  
**import** javafx.scene.control.Label;  
**import** javafx.scene.control.Button;  
**import** javafx.scene.control.CheckBox;  
**import** javafx.scene.control.TextField;  
**import** javafx.scene.control.TextArea;  
**import** javafx.scene.control.ComboBox;  
**import** javafx.scene.control.ScrollBar;  
**import** javafx.scene.layout.ColumnConstraints;  
**import** javafx.geometry.Orientation;  
**import** javafx.geometry.Pos;  
**import** javafx.geometry.VPos;  
  
//Create Main Class  
**public** **class** Main **extends** Application{   
//set variables  
Stage window;  
Scene scene;  
   
Label lb1;  
Label lb2;  
Label lb3;  
Label lb4;  
Label lb5;  
Label lb6;  
   
Button b1;  
Button b2;  
Button b3;  
   
ComboBox**<**String**>** comboBox1;  
ComboBox**<**String**>** comboBox2;  
ComboBox**<**String**>** comboBox3;  
   
//main method to launch  
**public** **static** **void** main(String[] args) {  
launch(args);  
}  
  
**@**Override  
**public** **void** start(Stage primaryStage) **throws** Exception{  
window **=** primaryStage;  
window.setTitle("Car Builder");  
   
   
//I used grid the get the layout I wanted  
GridPane grid **=** **new** GridPane();  
//The padding is to essentially set the margins for which my nodes won't go past  
grid.setPadding(**new** Insets(10,10,10,10));  
grid.setVgap(10);  
grid.setHgap(15);  
   
//label for Customer Name  
Label lb1 **=** **new** Label("Customer Name");  
GridPane.setConstraints(lb1, 0, 0);  
   
//label for Customer Name TextField  
TextField nameInput **=** **new** TextField();  
//I set a constraint as to how wide it should be because it was messing up the buttons look  
nameInput.setMaxWidth(200);  
GridPane.setConstraints(nameInput, 1, 0);  
   
//label for Car Make  
Label lb2 **=** **new** Label("Car Brand");  
GridPane.setConstraints(lb2, 0, 1);  
   
//label for Car Make combo box  
comboBox1 **=** **new** ComboBox**<>**();  
comboBox1.getItems().addAll("Cadillac $30,000", "Ford $13,000", "GM $15,000", "Honda $20,000", "Mercedes $45,000", "Nissan $14,000");  
GridPane.setConstraints(comboBox1, 1, 1);  
   
//label for Car Model  
Label lb3 **=** **new** Label("Car Model");  
GridPane.setConstraints(lb3, 0, 2);  
   
//label for Car Model combo box  
comboBox2 **=** **new** ComboBox**<>**();  
comboBox2.getItems().addAll("Convertible $14,250", "Coupe $12,000", "Hatchback $10,000", "Sedan $15,550", "SUV $25,750");  
GridPane.setConstraints(comboBox2, 1, 2);  
   
//label for Car Color  
Label lb4 **=** **new** Label("Car Color");  
GridPane.setConstraints(lb4, 0, 3);  
   
//label for Car Color combo box  
comboBox3 **=** **new** ComboBox**<>**();  
comboBox3.getItems().addAll("Black", "Blue", "Red", "Silver", "White");  
GridPane.setConstraints(comboBox3, 1, 3);  
  
//label for the check box Amenities Title  
Label lb5 **=** **new** Label("Amenities/Add-Ons");  
GridPane.setConstraints(lb5, 0, 4);  
   
//label for Check Box 1  
CheckBox c1 **=** **new** CheckBox("Air Conditioned Seats $750");  
GridPane.setConstraints(c1, 1, 4);  
   
//label for Check Box 2  
CheckBox c2 **=** **new** CheckBox("Automatic Emergency Braking $5000");  
GridPane.setConstraints(c2, 2, 4);  
   
//label for Check Box 3  
CheckBox c3 **=** **new** CheckBox("Backup Camera $550");  
GridPane.setConstraints(c3, 1, 5);  
   
//label for Check Box 4  
CheckBox c4 **=** **new** CheckBox("Blue-Tooth Connectivity $550");  
GridPane.setConstraints(c4, 2, 5);  
   
//label for Check Box 5  
CheckBox c5 **=** **new** CheckBox("Heated Seats $750");  
GridPane.setConstraints(c5, 1, 6);  
   
//label for Check Box 6  
CheckBox c6 **=** **new** CheckBox("Keyless Entry $650");  
GridPane.setConstraints(c6, 2, 6);  
   
//label for Check Box 7  
CheckBox c7 **=** **new** CheckBox("Navigation System $1100");  
GridPane.setConstraints(c7, 1, 7);  
   
//label for Check Box 8  
CheckBox c8 **=** **new** CheckBox("Super Sound System $1,200");  
GridPane.setConstraints(c8, 2, 7);  
   
//label and text box for output   
Label lb6 **=** **new** Label("Car Specs and Price");  
GridPane.setConstraints(lb6, 0, 8);  
   
TextArea Out **=** **new** TextArea();  
//I set a constraint as to how wide it should be because it was messing up the buttons look  
Out.setMaxWidth(300);  
GridPane.setConstraints(Out, 1, 8);

//label for button1  
Button b1 **=** **new** Button("Close Program");  
GridPane.setConstraints(b1, 0, 9);  
//using lambda I shortened the expression to close when the button is pressed  
b1.setOnAction(e **->** System.exit(0));   
//label for button2  
//Button 2 clears the programd so you can start a new build  
Button b2 **=** **new** Button("Clear");  
GridPane.setConstraints(b2, 1, 9);  
//This is my manual reset of all the fields  
b2.setOnAction(e **->** {  
//set the nameInput box to an empty string  
nameInput.setText("");  
//set the comboBoxes to null to clear the selection  
comboBox1.setValue(**null**);  
comboBox2.setValue(**null**);

comboBox3.setValue(**null**);  
//set all the checkboxes to false  
c1.setSelected(**false**);  
c2.setSelected(**false**);  
c3.setSelected(**false**);  
c4.setSelected(**false**);  
c5.setSelected(**false**);  
c6.setSelected(**false**);  
c7.setSelected(**false**);  
c8.setSelected(**false**);  
//set the Output TextArea to an empty string  
Out.setText("");  
});//end button action   
//label for button3  
Button b3 **=** **new** Button("Calculate Car costs");  
GridPane.setConstraints(b3, 2, 9);  
b3.setOnAction(e **->** {  
//Create an empty string for the Out TextArea  
String output **=** "";  
//Cost is for when we add the values together  
**int** cost **=** 0;  
//First line takes the nameInput and puts it in output  
output **=** "Customer name : " **+** nameInput.getText()**+**"\n";  
//make takes the value from the Car Brand Check Box and assigns it to cmake  
String brand**=** "" **+** comboBox1.getValue();  
//Takes the chosen car and adds the string to output and the adds the price to cost  
**if**(brand.equals("Cadillac $30,000")){  
output **=** output **+** "Car Brand and Price: Cadillac $30,000\n";  
cost **=** cost **+** 30000;   
}//end if  
//Takes the chosen car and adds the string to output and the adds the price to cost  
**else** **if**(brand.equals("Ford $13,000")){  
output **=** output **+** "Car Brand and Price: Ford $13,000\n";  
cost **=** cost **+** 13000;  
}//end else if  
//Takes the chosen car and adds the string to output and the adds the price to cost  
**else** **if**(brand.equals("GM $15,000")){  
output **=** output **+** "Car Brand and Price: GM $15,000\n";  
cost **=** cost **+** 15000;  
}//end else if  
//Takes the chosen car and adds the string to output and the adds the price to cost  
**else** **if**(brand.equals("Honda $20,000")){  
output **=** output **+** "Car Brand and Price: Honda $20,000\n";  
cost **=** cost **+** 20000;  
}//end else if  
//Takes the chosen car and adds the string to output and the adds the price to cost  
**else** **if**(brand.equals("Mercedes $45,000")){  
output **=** output **+** "Car Brand and Price: Mercedes $45,000\n";  
cost **=** cost **+** 45000;  
} //end else if   
//If no chosen car brand it defaults to Nissan and adds the string to output and the adds the price to cost  
**else**{  
output **=** output **+** "Car Brand and Price: Nissan $14,000\n";  
cost **=** cost **+** 14000;   
}//end else  
//creates a String model to hold information about which car model was chosen  
String model**=** "" **+** comboBox2.getValue();  
   
//Takes the chosen model and adds the string to output and the adds the price to cost  
**if**(model.equals("Convertible $14,250")){  
output **=** output **+** "Model: Convertible $14,250\n";  
cost **=** cost **+** 14250;  
}//end if  
//Takes the chosen model and adds the string to output and the adds the price to cost  
**else** **if**(model.equals("Coupe $12,000")){  
output **=** output **+** "Model: Coupe $12,000\n";  
cost **=** cost **+** 12000;  
}//end else if   
//Takes the chosen model and adds the string to output and the adds the price to cost  
**else** **if**(model.equals("Hatchback $10,000")){  
output **=** output **+** "Model: Hatchback $10,000\n";  
cost **=** cost **+** 10000;   
}//end else if  
//Takes the chosen model and adds the string to output and the adds the price to cost  
**else** **if**(model.equals("Sedan $15,550")){  
output **=** output **+** "Model: Sedan $15,550\n";  
cost **=** cost **+** 15550;  
}//end else if   
//If no chosen model it defaults to SUV and adds the string to output and the adds the price to cost  
**else**{  
output **=** output **+** "Model: SUV $25,750\n";  
cost **=** cost **+** 25750;   
}//end else if  
//Takes the chosen color and adds the string to output  
String color**=** "" **+** comboBox3.getValue();  
**if**(color.equals("Black")){  
output **=** output **+** "Car color: Black\n";  
}//end if  
//Takes the chosen color and adds the string to output  
**else** **if**(color.equals("Blue")){  
output **=** output **+** "Car color: Blue\n";  
}//end else if  
//Takes the chosen color and adds the string to output  
**else** **if**(color.equals("Red")){  
output **=** output **+** "Car color: Red\n";  
}//end else if  
//Takes the chosen color and adds the string to output  
**else** **if**(color.equals("Silver")){  
output **=** output **+**"Car color: Silver\n";  
}//end else if  
//If no chosen color it will default to white and add the string to output  
**else**{   
output **=** output **+** "Car color: White\n";  
}//end else   
//if the checkbox is selected the system will add the price to the cost and the output string   
**if**(c1.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Air Conditioned Seats $750\n";  
cost **=** cost **+** 750;  
}//end if  
//if the checkbox is selected the system will add the price to the cost and the output string  
**if**(c2.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Automatic Emergency Braking $5000\n";  
cost **=** cost**+**5000;  
}//end if   
//if the checkbox is selected the system will add the price to the cost and the output string  
**if**(c3.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Backup Camera $550\n";  
cost **=** cost **+** 550;  
}//end if  
//if the checkbox is selected the system will add the price to the cost and the output string  
**if**(c4.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Blue-Tooth Connectivity $550\n";  
cost **=** cost **+** 50;  
}//end if  
//if the checkbox is selected the system will add the price to the cost and the output string  
**if**(c5.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Heated Seats $750\n";  
cost **=** cost **+** 750;  
}//end if  
//if the checkbox is selected the system will add the price to the cost and the output string  
**if**(c6.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Keyless Entry $650\n";  
cost **=** cost **+** 650;  
}//end if  
 //if the checkbox is selected the system will add the price to the cost and the output string  
**if**(c7.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Navigation System $1100\n";  
cost **=** cost **+** 1100;  
}//end if  
//if the checkbox is selected the system will add the price to the cost and the output string  
**if**(c8.isSelected()){  
output **=** output **+** "Added Amenities/Add-Ons: Super Sound System $1,200\n";  
cost**=**cost **+** 1200;  
}//end if  
//Adds the Total Cost to the Output string  
output **=** output **+** "The Total Price of the car will be: $" **+** cost;  
//places the output string in the Out TextArea  
Out.setText(output);  
});//end button action   
   
//combining them all together using getChildren  
grid.getChildren().addAll(nameInput, lb1, lb2, lb3, lb4, lb5, lb6, b1, b2,  
b3, comboBox1, comboBox2, comboBox3, Out, c1,c2,c3,c4,c5,c6,c7,c8);  
  
//set scene parameter to grid as the layout profile   
scene **=** **new** Scene(grid, 800, 800);  
//set scene  
window.setScene(scene);  
//show display the stage  
window.show();  
}//end method  
}//end class

**)4.2 Program and Module Description**

Main

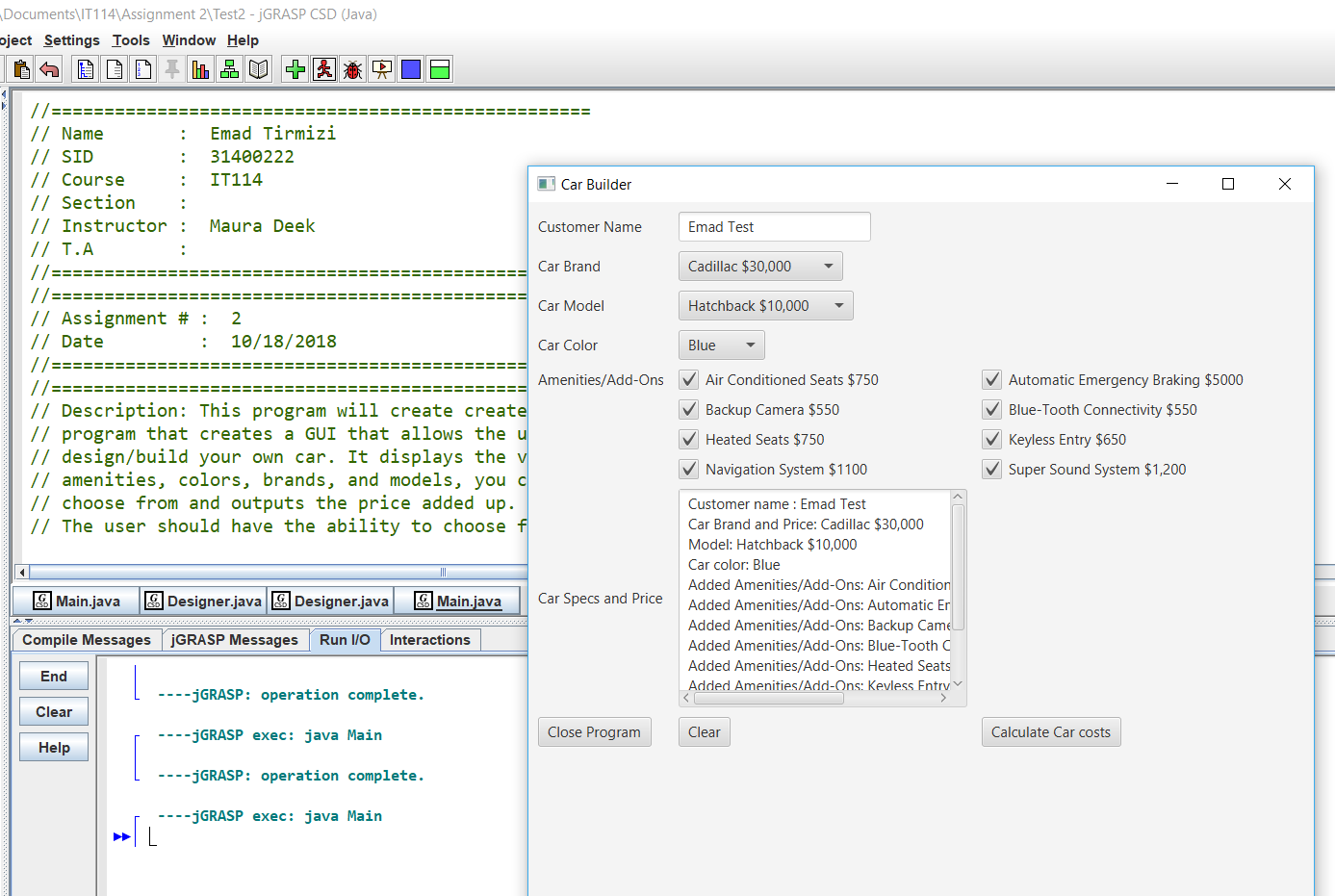
The main function creates the program GUI and holds all the information regarding car brands, colors, models, and amenities. It also carries out the function of calculating and clearing the program, as well as exiting from the program.

**5. Solution Testing**

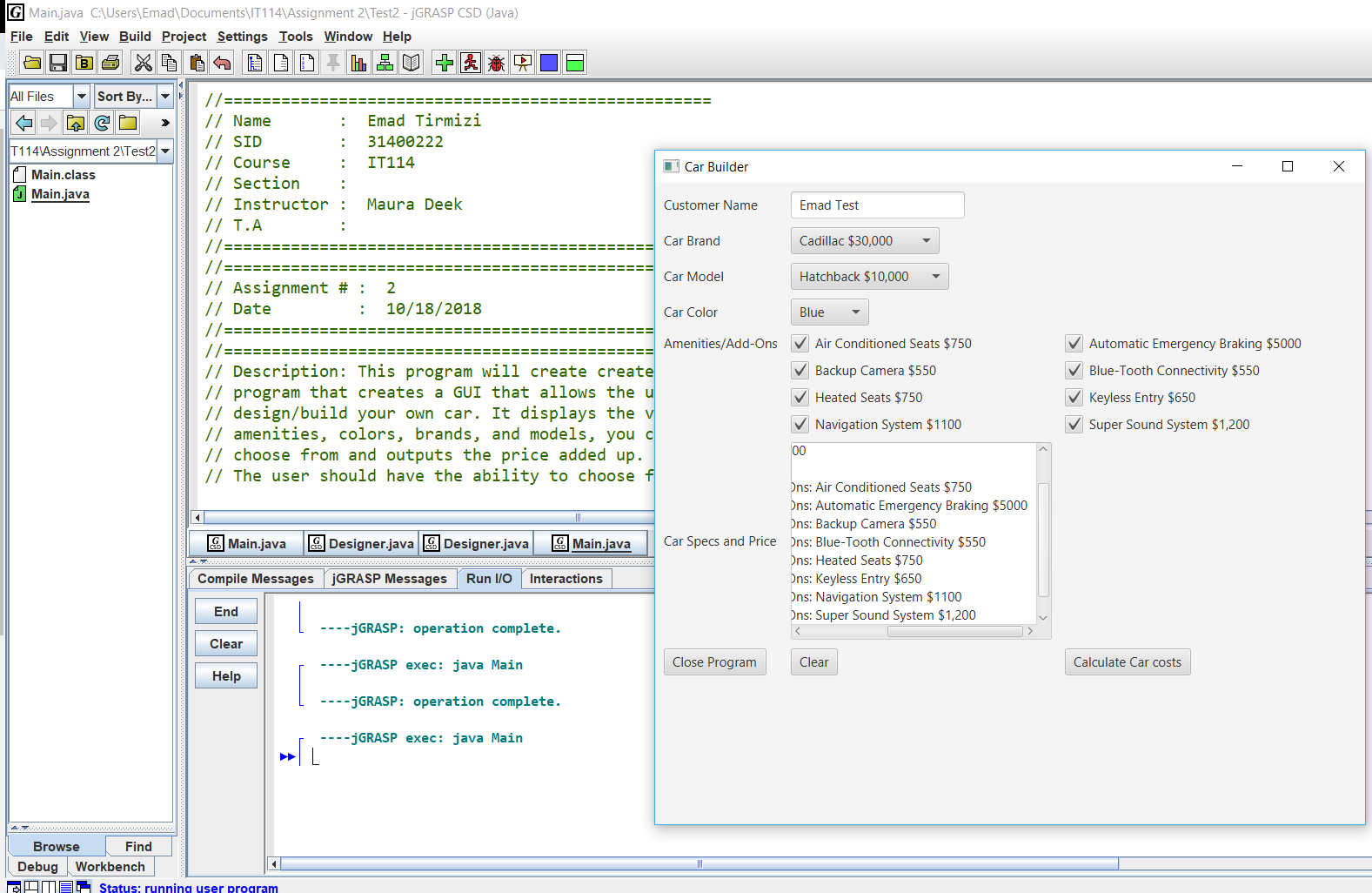
Test the program with following data domain:

Any Combination

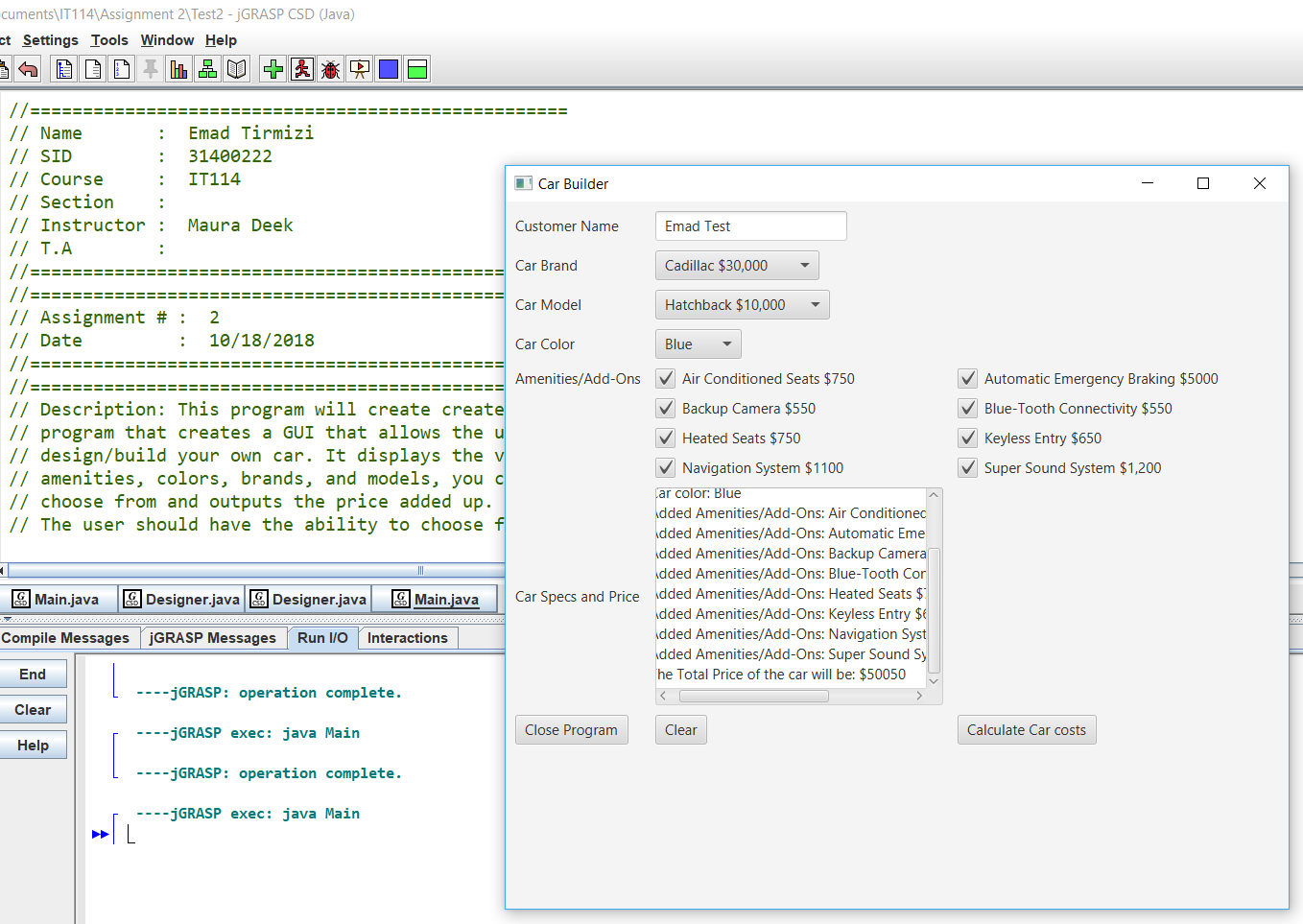
These are some runs that I did with different Variables.



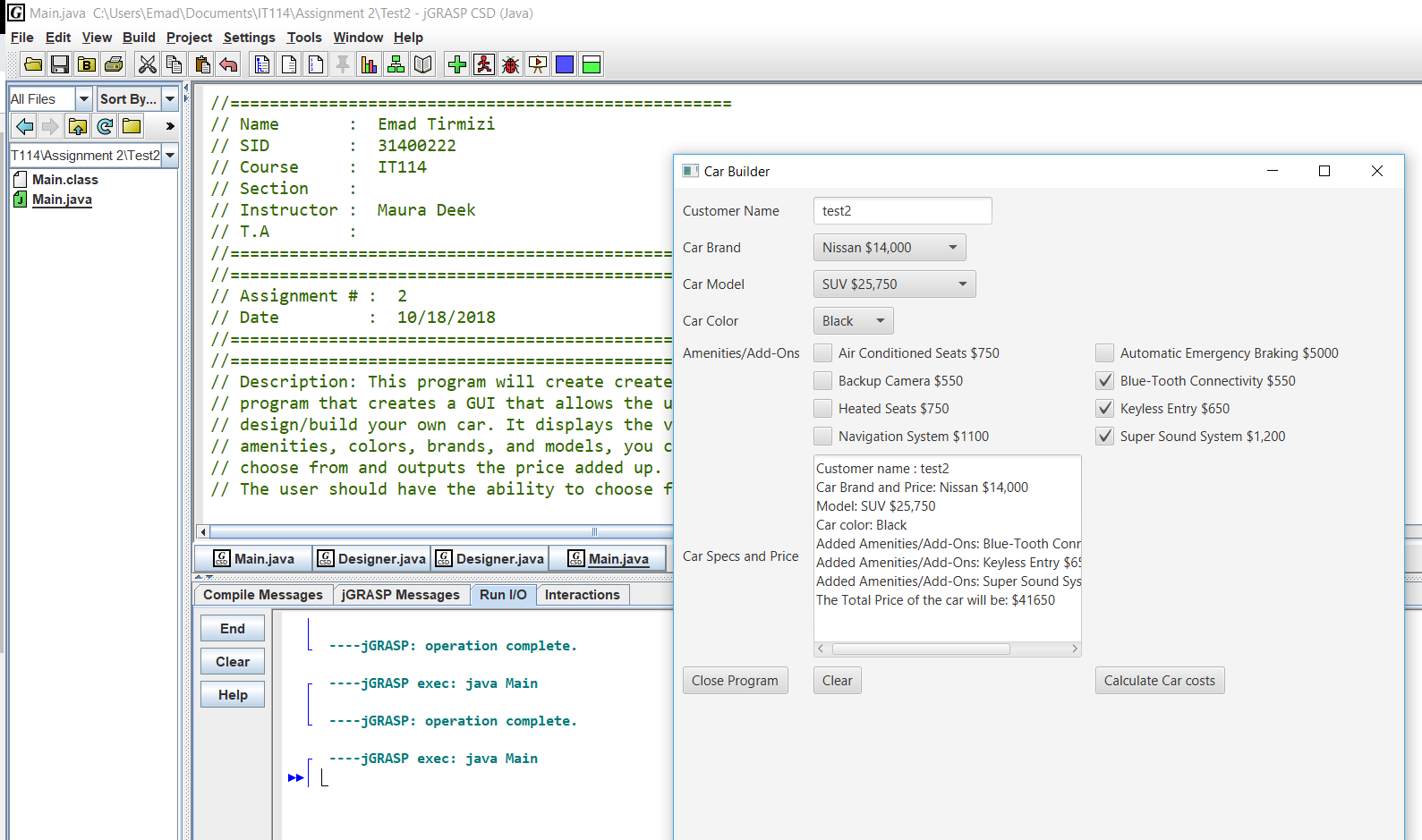
This one is with the same variables, However I wanted to show you the rest of the output



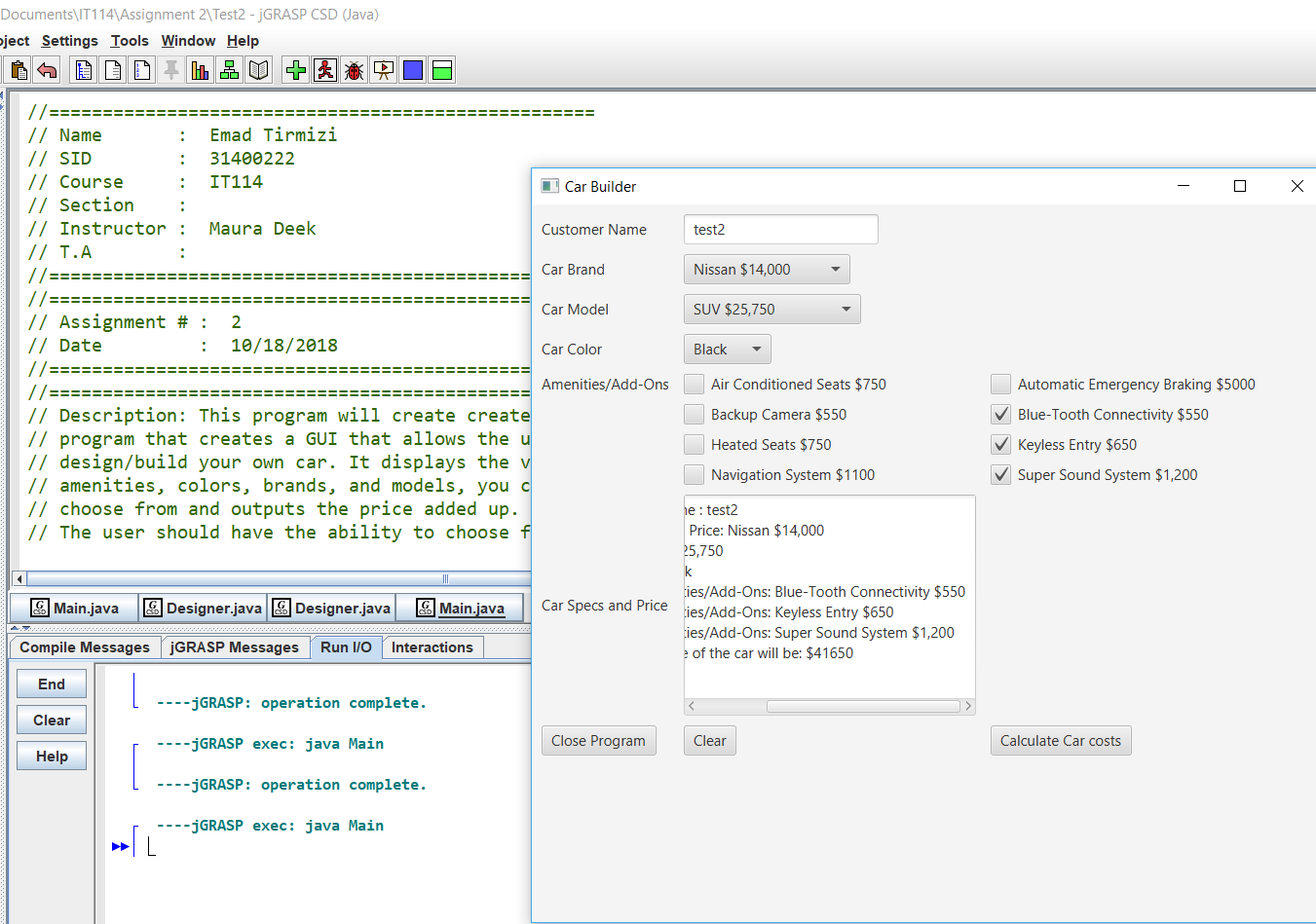
This one is the same as the previous, except this one shows the total cost at the bottom of the TextArea



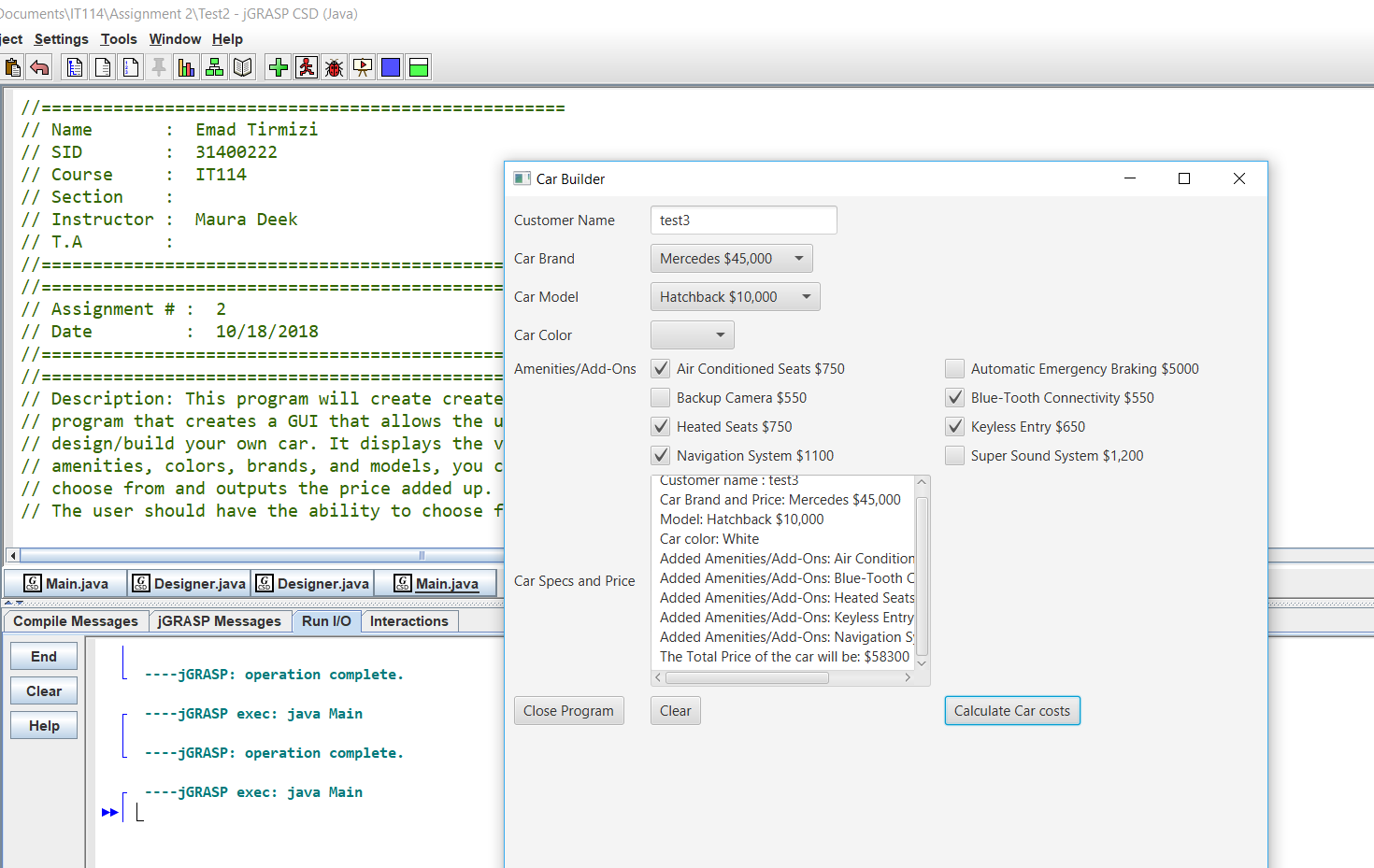
This is a different set up I ran



The rest of the Output of the previous picture



New variables I ran.



Continued output from the last picture showing the rest of the output

