CISP 41

Programming in C#

Project Evaluation Sheet

Student Name: _____ Bryant Tunbutr _____ Project Number: _____3_____

Project Name: _____BtunbutrProject3_____ Visual Studio Version:___2008_____

Date Due: _____11/1/12_____ Date Turned In: _____11/1/12_____

Above to be completed by student

Points (40 Possible)

Correctness/Efficiency:

Output is accurate _____

Meets all requirements Arrays are an important part of this project. -5

Provide appropriate user interface _____

Logic is efficient _____

Documentation/Coding Style:

Project can be open from the submitted zip file _____

Folder is present and contains all necessary project files (no extra files) _____

Use required coding template _____

Use proper naming and spacing _____

Submit all requested information _____

Test Cases:

List all required test cases _____

Provide output forms for important test cases _____

Other issues: _____

Extra Credit: _____

Timeliness: _____

Project Score: 35/40

CISP 41 Programming in C#

Project specification

This software is intended to summarize, calculate, and display costs for repairs for Bryant's Auto Repair Shop.

It is designed to be run in Visual Studio 2008 using the C# coding.

It uses user input including the selected part to be repaired and number of hours.

It displays total cost, summary, and provides information about the program with an about button.

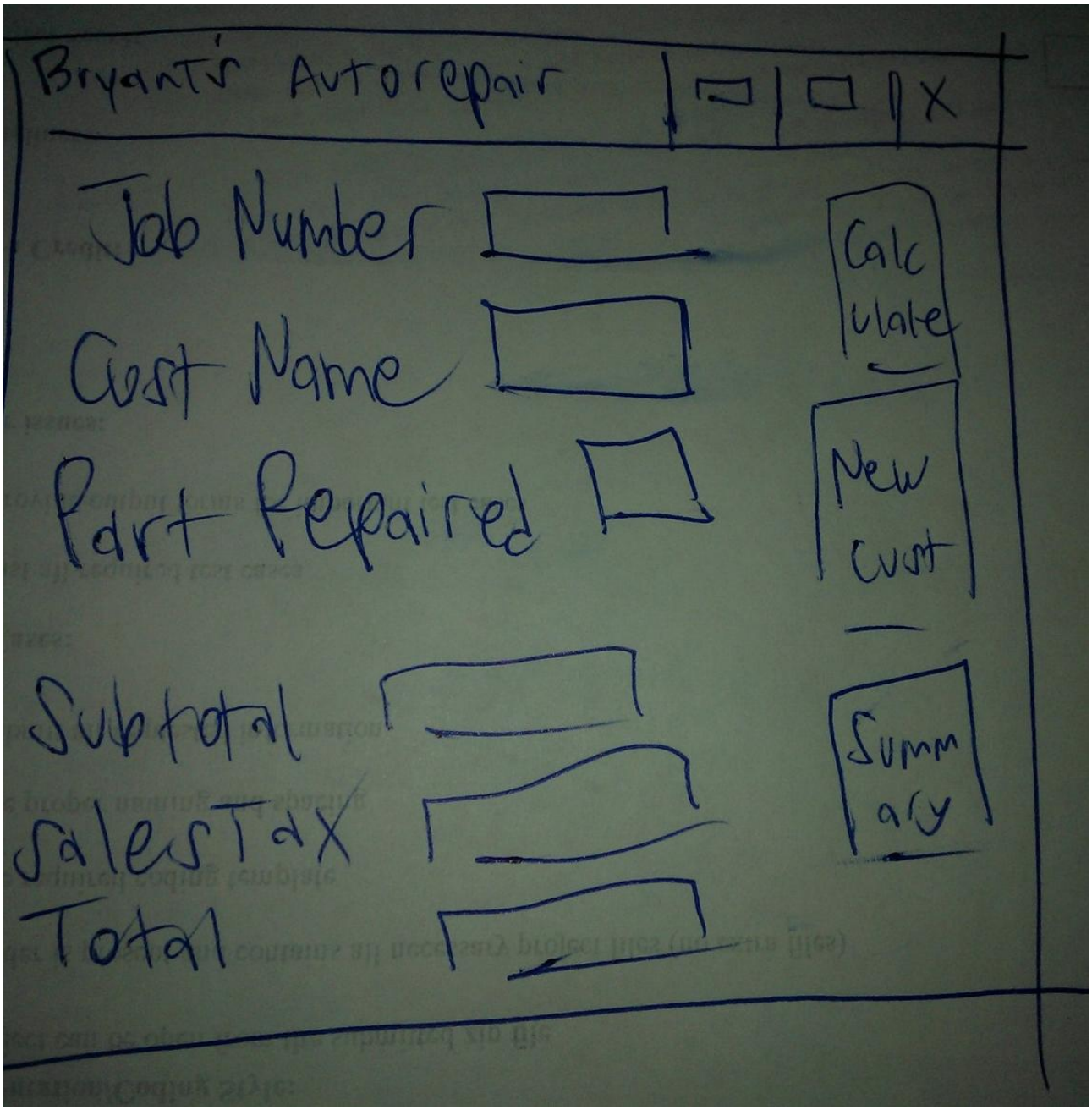
Used in the project are arrays, methods, calculations, multiple forms including summary and about and splash.

Project status

The project is completed and finalized..

CISP 41 **Programming in C#**

Sketch of user interface



CISP 41 **Programming in C#**

Objects and Properties Plan for __ BTunbutrProject 3_____Form

Object	Property	Setting
exitButton	Name Text	**E&xit**
New Customer Button	Name Text	**&New Customer**
Subtotal	Text	**Method**
label	Name Text	**Summary**
Combo text box	Text	**Drop down**
about	Name Text	**Info**
BTunbutrProject3	Name Text	**Form**
Menu	Name Text	**Click**
splash	Text	**Form**

CISP 41 **Programming in C#**

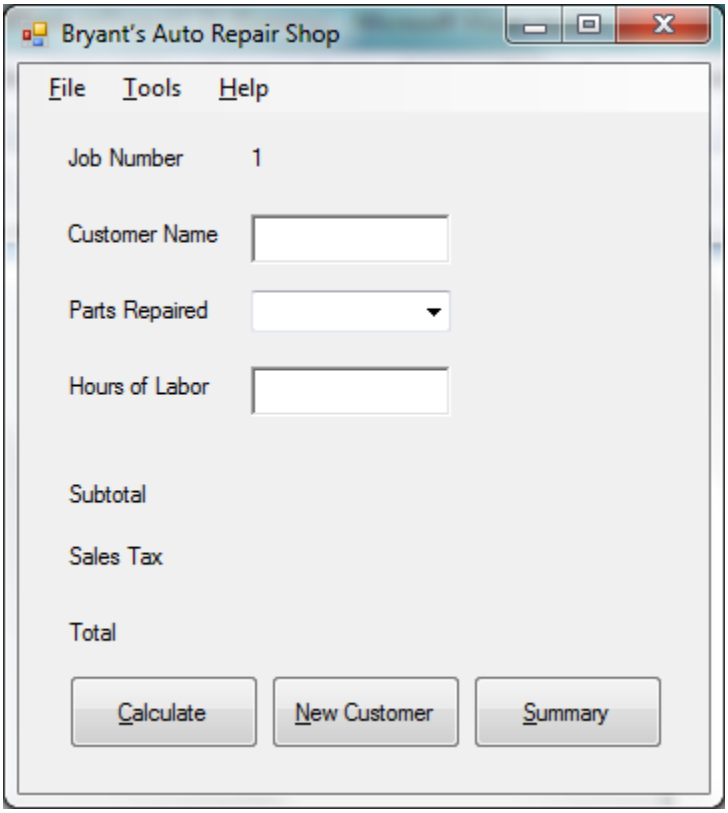
Event Plan for ___ BTunbutrProject 3_____Form

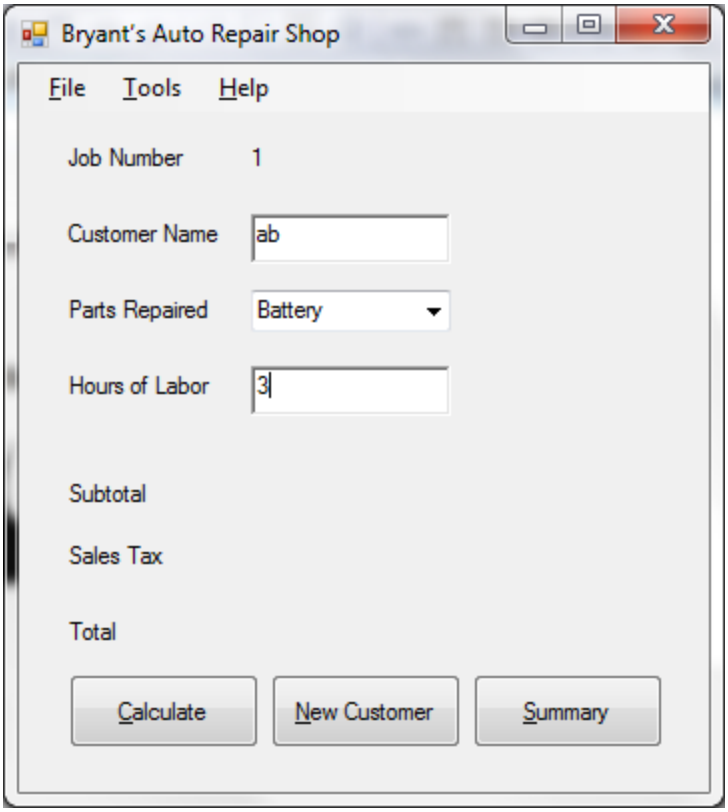
Object	Event	Action - Pseudocode
summaryButton	Click	Call summaryButton_Click. Display the price of the order. Put new form up Add to the number of orders and totalSales. Reset the controls for the next order. Clear the order amount
About	Click	Display the program info in a new form
aboutToolStripMenuItem	Click	Display a message box showing programmer and version
exitButton	Click	exit
Menu	Click	Buttons that are the same name
Drop down	Click	Select array

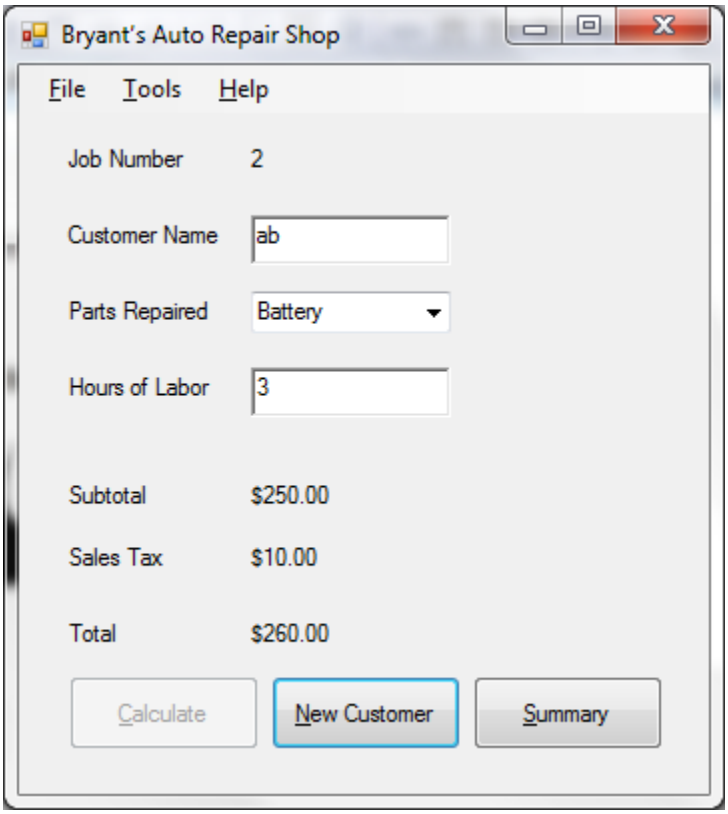
CISP 41 Programming in C#

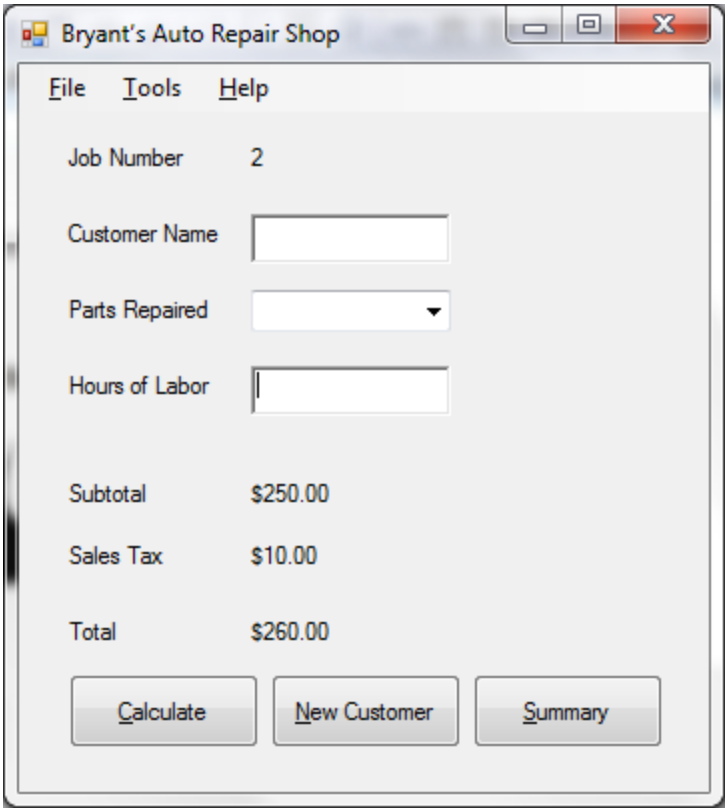
Test cases and captured screens

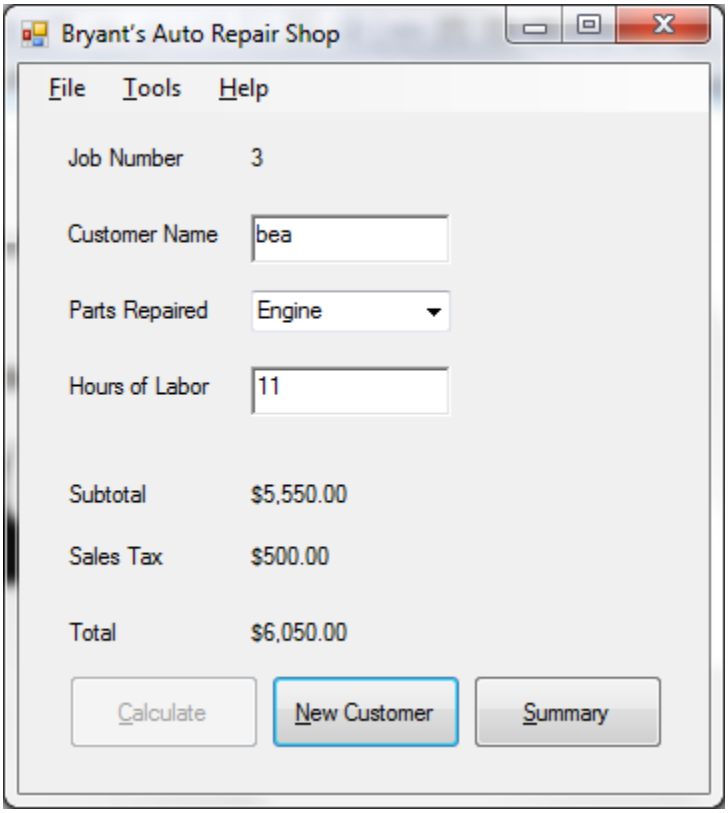
Test case #1
Ab orders battery with 3 hours of labor
Bea orders engine with 11 hours of labor

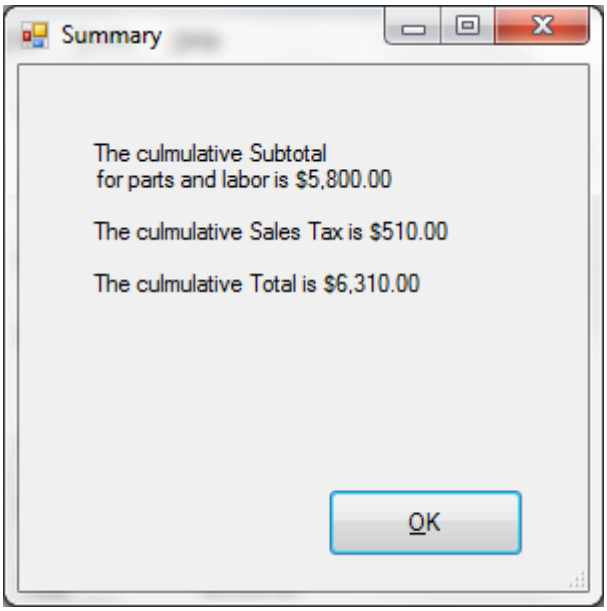



Bryant's Auto Repair Shop
File
Tools
Help
Job Number
1
Customer Name
ab
Parts Repaired
Battery
Hours of Labor
3
Subtotal
Sales Tax
Total
Calculate
New Customer
Summary

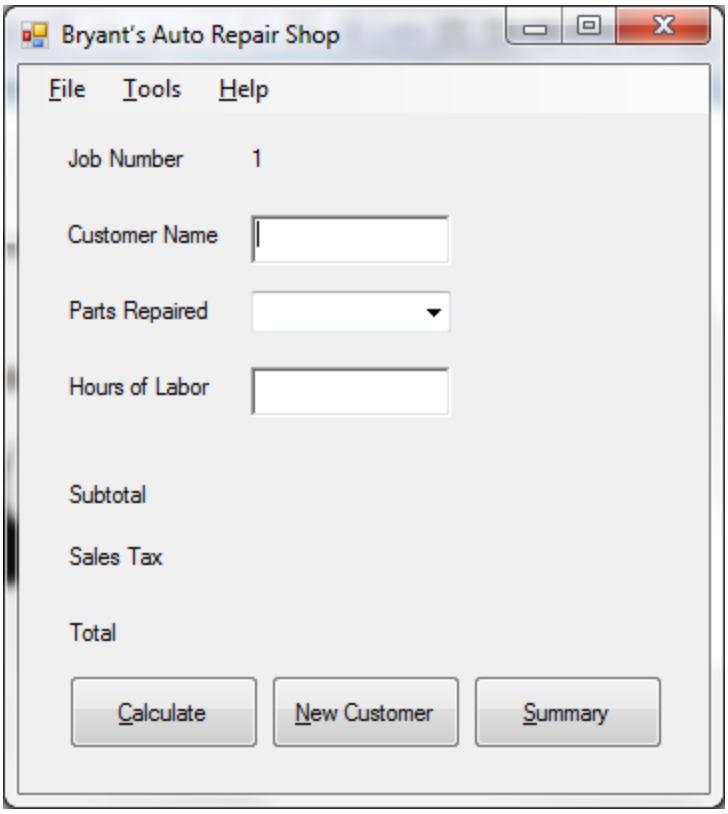

Bryant's Auto Repair Shop
File
Tools
Help
Job Number
2
Customer Name
ab
Parts Repaired
Battery
Hours of Labor
3
Subtotal
$250.00
Sales Tax
$10.00
Total
$260.00
Calculate
New Customer
Summary

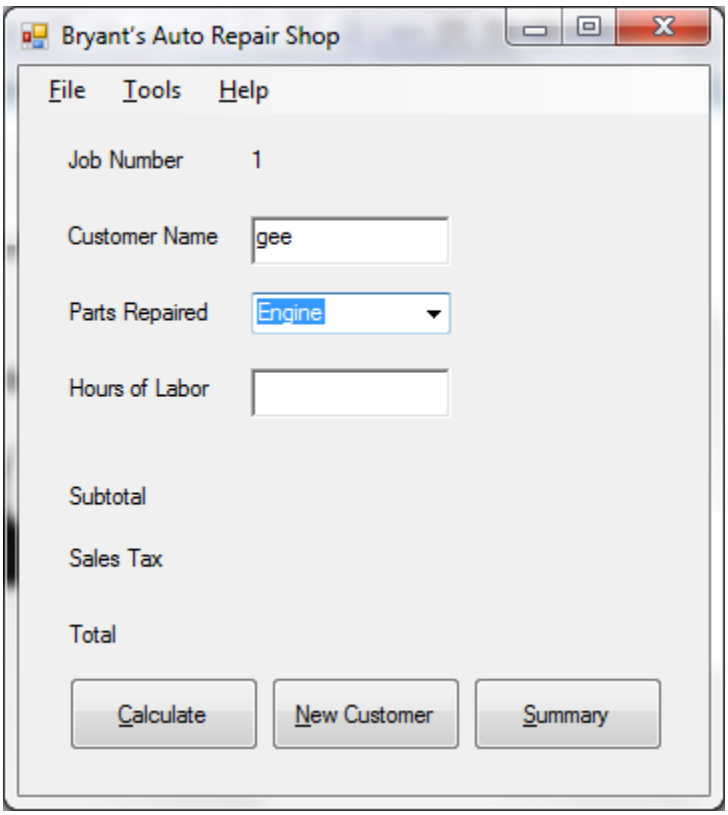

Bryant's Auto Repair Shop
File
Tools
Help
Job Number
2
Customer Name
Parts Repaired
Hours of Labor
Subtotal
$250.00
Sales Tax
$10.00
Total
$260.00
Calculate
New Customer
Summary

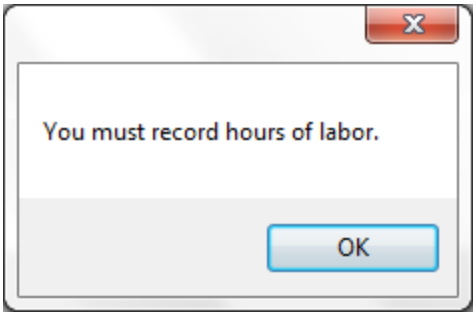

Bryant's Auto Repair Shop
File
Tools
Help
Job Number
3
Customer Name
bea
Parts Repaired
Engine
Hours of Labor
11
Subtotal
$5,550.00
Sales Tax
$500.00
Total
$6,050.00
Calculate
New Customer
Summary

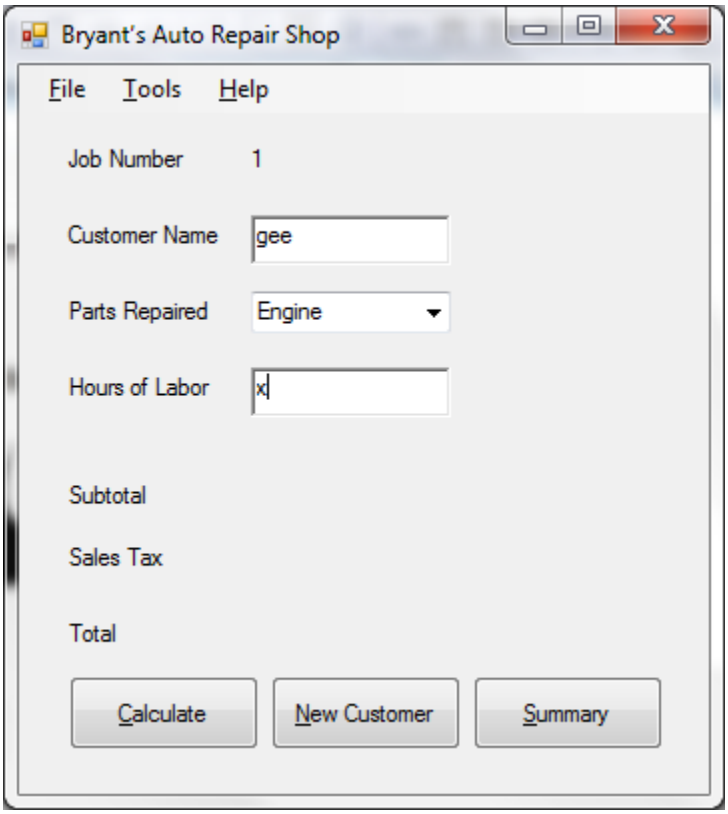

Summary
The culmulative Subtotal
for parts and labor is $5,800.00
The culmulative Sales Tax is $510.00
The culmulative Total is $6,310.00
OK

Test case #2
Missing and wrong data input

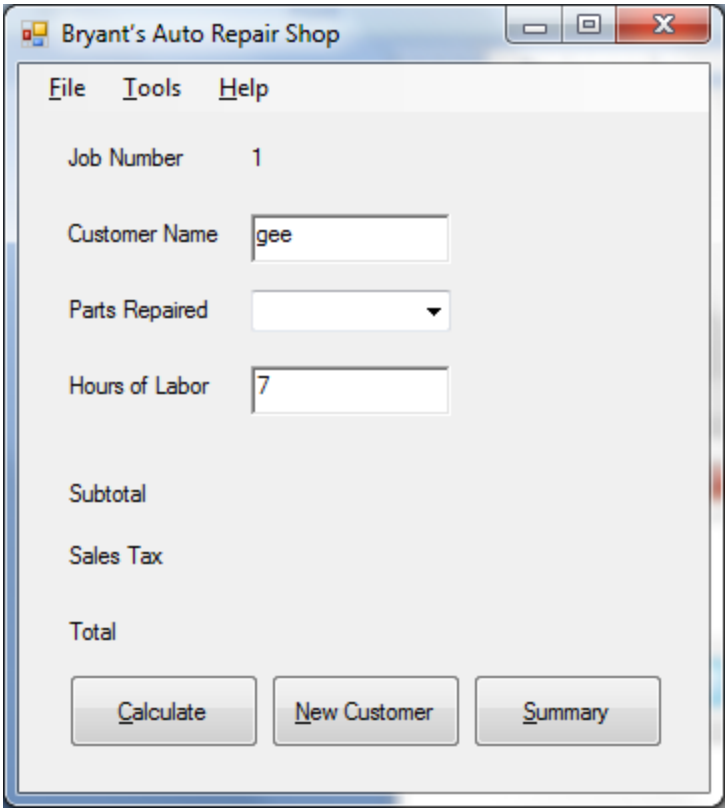


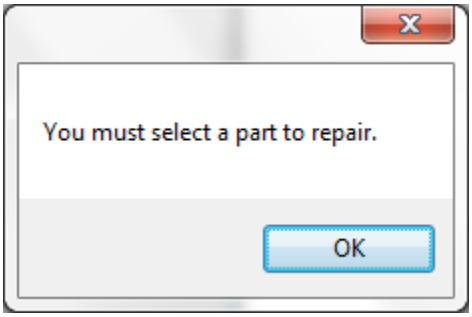



You must record hours of labor.
OK

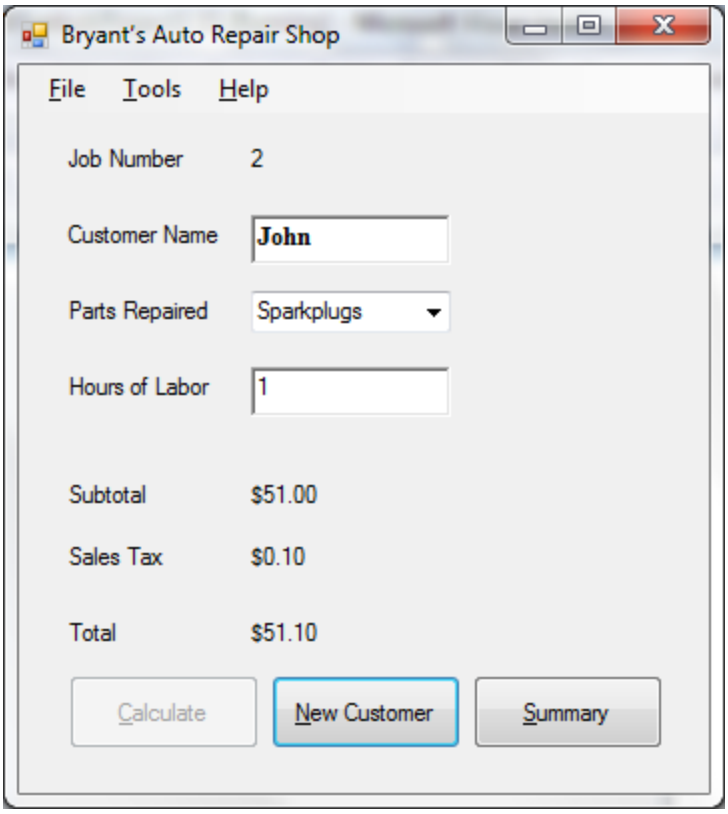

Bryant's Auto Repair Shop
File
Tools
Help
Job Number
1
Customer Name
gee
Parts Repaired
Engine
Hours of Labor
x
Subtotal
Sales Tax
Total
Calculate
New Customer
Summary

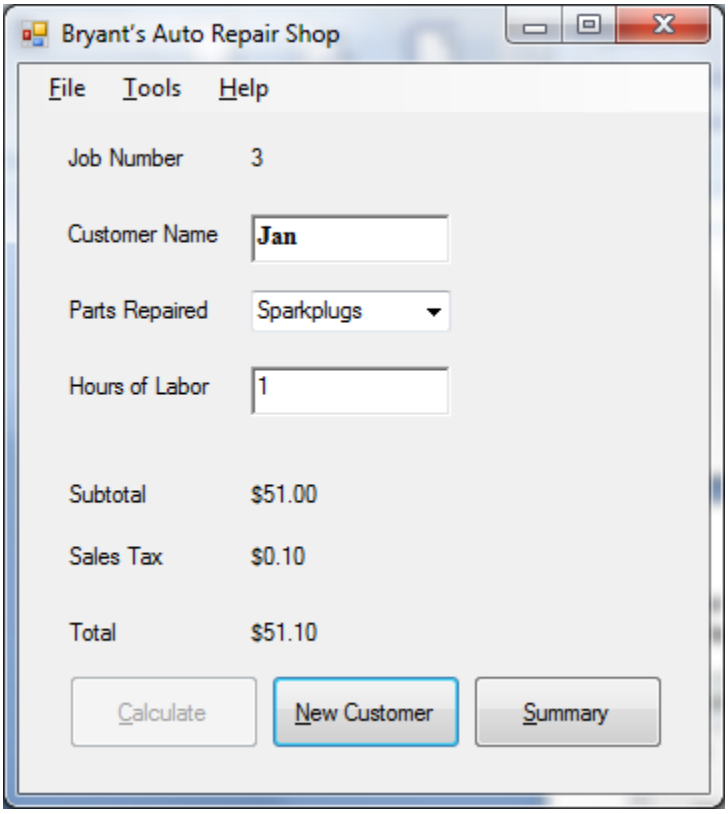

Data Entry Error
Invalid quantity.
OK

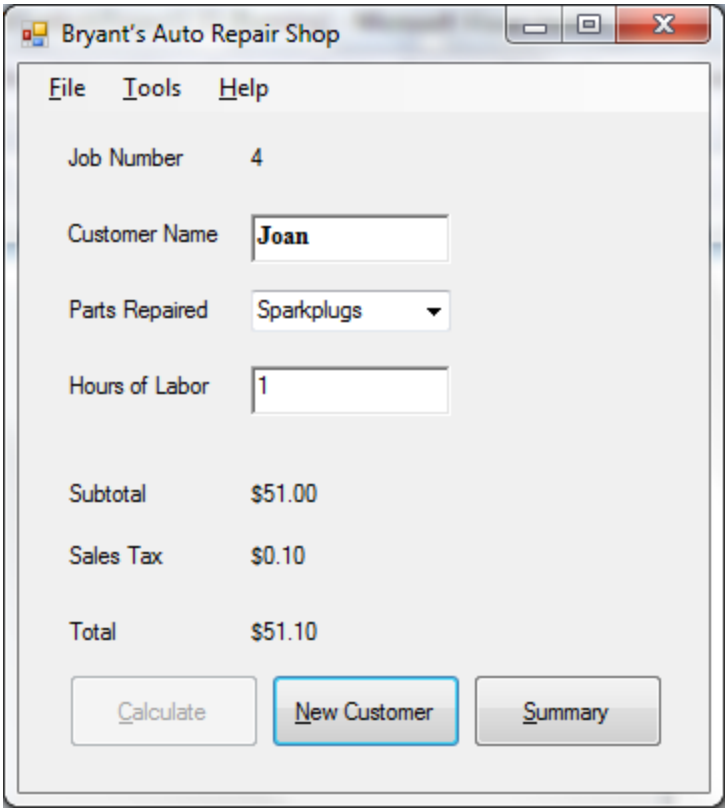

Bryant's Auto Repair Shop
File
Tools
Help
Job Number
1
Customer Name
gee
Parts Repaired
Hours of Labor
7
Subtotal
Sales Tax
Total
Calculate
New Customer
Summary

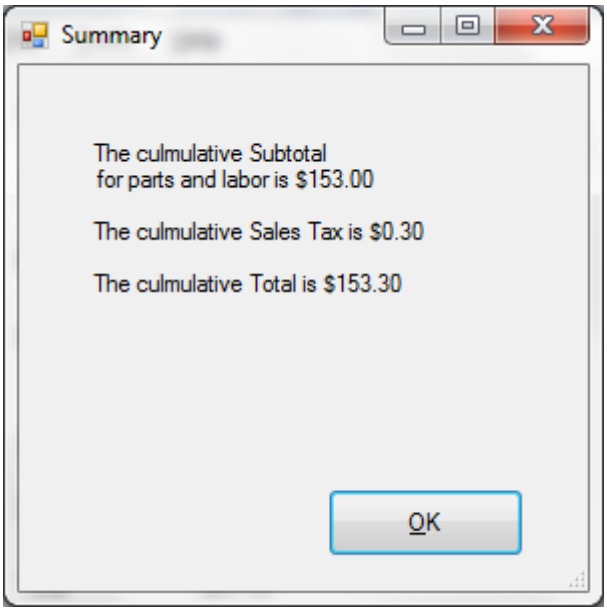

You must select a part to repair.
OK

Test case #3
John, Jan, Joan each buy spark plug with 1 hour of labor each

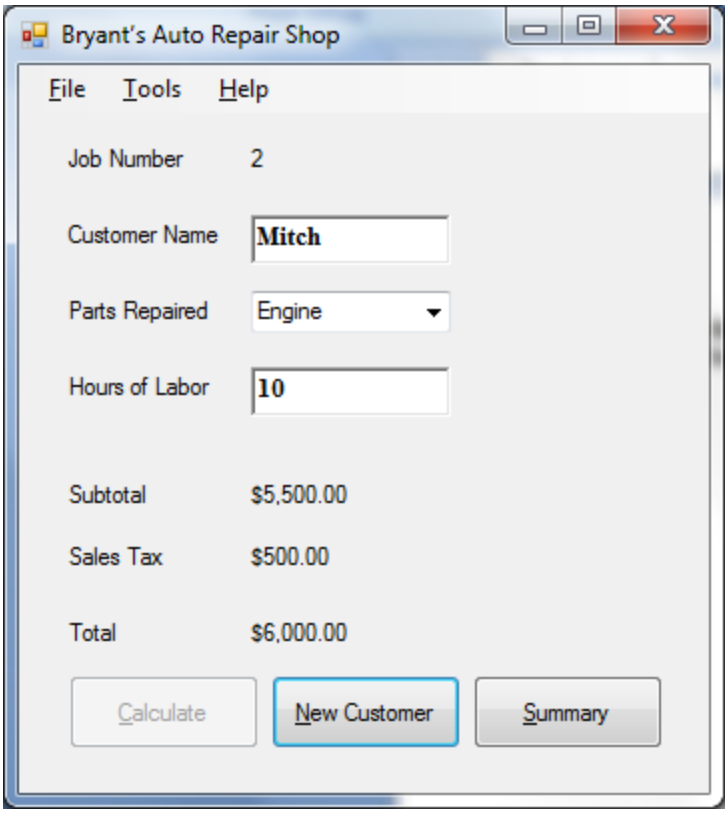


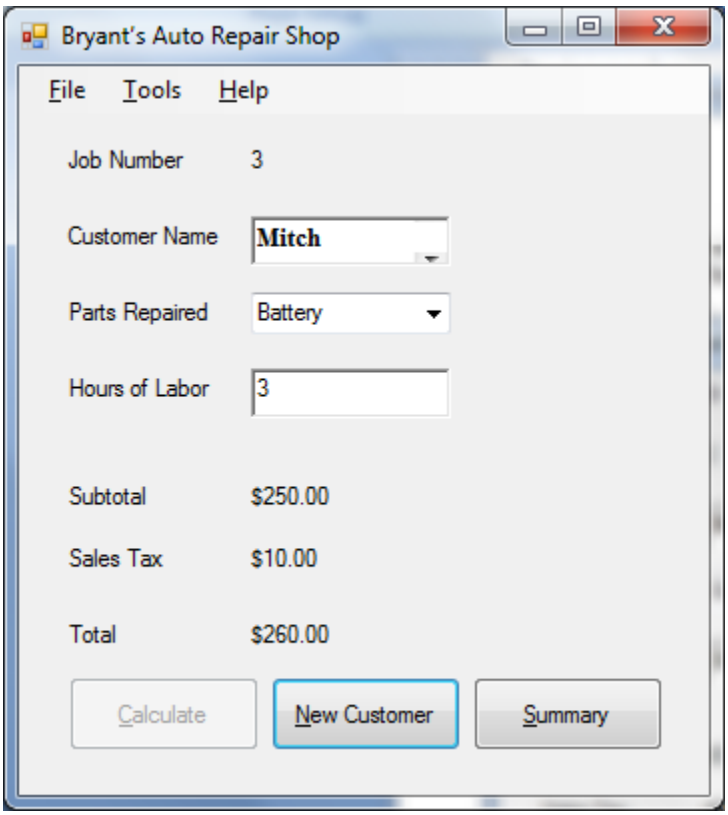


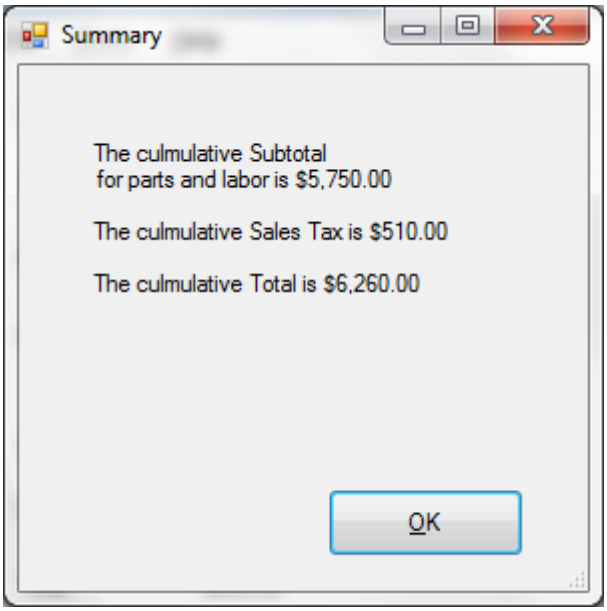

Bryant's Auto Repair Shop
File
Tools
Help
Job Number
4
Customer Name
Joan
Parts Repaired
Sparkplugs
Hours of Labor
1
Subtotal
$51.00
Sales Tax
$0.10
Total
$51.10
Calculate
New Customer
Summary


Summary
The culmulative Subtotal
for parts and labor is $153.00
The culmulative Sales Tax is $0.30
The culmulative Total is $153.30
OK

Test case #4
Mitch buys an engine that takes 10 hours
Then comes back and buys a battery that takes 3 hours






Summary
The culmulative Subtotal
for parts and labor is $5,750.00
The culmulative Sales Tax is $510.00
The culmulative Total is $6,260.00
OK

Main form

```
/*
* Project: BtunbutrProject3
* Programmer: Bryant Tunbutr
* Date: Nov 1 2012
* Description: Calculates and displays the cost for autoshop materials and labor
 * I certify that the code below is my own work.
*/

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;


namespace WindowsFormsApplication14
{

    public partial class Form1 : Form
    {
        public struct GroupInfo
        {
            //public string groupNumberString;
            public decimal totalDecimal, groupNumberDecimal, totalsDecimal;
            public int groupNumberInteger;
        }

        // Declare the variables.
        decimal partsChargesDec, subTotalDec,
            salesTaxDec, totalDec, laborChargesDec,
            summarySalesTaxDec, summarySubTotalDec, summaryTotalDec;
        int number, groupNumberInteger;

        public Form1()
        {
            InitializeComponent();
            //Intialize the variable to 1
            number = 1;
            //Intialize the label to 1 as well
            jobLabel.Text = number.ToString();
        }

        private void calculateToolStripMenuItem_Click(object sender, EventArgs
e)
        {
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
```

```
            this.Close();
        }

        private void newCustomerToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            // clear boxes
            customerNameTextBox.Text = "";
            hoursOfLaborTextBox.Text = "";
            groupComboBox.Text = "";

            //set focus
            hoursOfLaborTextBox.Focus();

            //enable calculate
            calculateButton.Enabled = true;
        }

        private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //show about form
            AboutBox1 aboutForm = new AboutBox1();
            aboutForm.ShowDialog();
        }

        private void jobLabel_Click(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Add the current item price and quantity to the order.


            //check for missing info
            if (groupComboBox.SelectedIndex == -1)
            {
                MessageBox.Show("You must select a part to repair.");
            }
            if (hoursOfLaborTextBox.Text == "")
            {
                MessageBox.Show("You must record hours of labor.");
            }
            else
            {
                try
                {
                    //calculate by using what customer selected, match with
prices here
                    groupNumberInteger = groupComboBox.SelectedIndex;
                    switch (groupComboBox.SelectedIndex)
                    {
                        case 0:
                            partsChargesDec = 100m;
                            break;
                        case 1:
                            partsChargesDec = 5000m;
                            break;
                        case 2:
                            partsChargesDec = 1m;
                            break;
```

Must use arrays to store and look up information as specified.

```
                        case 3:
                            partsChargesDec = 50m;
                            break;
                    }

                    decimal hoursOfLaborDec =
decimal.Parse(hoursOfLaborTextBox.Text);

                    // Calculate values.
                    salesTaxDec = partsChargesDec * .1m;

                    laborChargesDec = hoursOfLaborDec * 50m;

                    subTotalDec = partsChargesDec + laborChargesDec;

                    totalDec = subTotalDec + salesTaxDec;

                    //Display the results
                    salesTaxInfoLabel.Text = salesTaxDec.ToString("C");
                    subtotalInfoLabel.Text = subTotalDec.ToString("C");
                    totalInfoLabel.Text = totalDec.ToString("C");

                    //Increase job number, increment the number
                    number++;
                    //Update the label. Convert the number to a string
                    jobLabel.Text = number.ToString();

                    //disable calculate
                    calculateButton.Enabled = false;

                    // running total of sales tax, subtotal, total for all
customers
                    summarySalesTaxDec += salesTaxDec;
                    summarySubTotalDec += subTotalDec;
                    summaryTotalDec += totalDec;
                }
                catch (FormatException)
                {
                    // Error message for bad info.
                    MessageBox.Show("Invalid quantity.", "Data Entry Error");
                }
            }
        }

        private void summaryButton_Click(object sender, EventArgs e)
        {
            // Display the summary form.
            SummaryForm aSummaryForm = new SummaryForm();

            // send info to the summary form.
            aSummaryForm.SummarySubTotal = summarySubTotalDec;
            aSummaryForm.SummarySalesTax = summarySalesTaxDec;
            aSummaryForm.SummaryTotal = summaryTotalDec;

            // make modal summary form.
            aSummaryForm.ShowDialog();
              }

    }
}
```

Summary form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication14
{
    public partial class SummaryForm : Form
    {
        // make values accessible to this summary form

        decimal summarySubTotalDec,summarySalesTaxDec,summaryTotalDec;

        public decimal SummarySubTotal
        {
            set
            {
            summarySubTotalDec = value;
            }
        }

        public decimal SummarySalesTax
        {
            set
            {
                summarySalesTaxDec = value;
            }
        }
        public decimal SummaryTotal
        {
            set
            {
                summaryTotalDec = value;
            }
        }

        public SummaryForm()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            // display summary
            summaryLabel.Text = "The culmulative Subtotal" + "\r\n" + " for
parts and labor is " +
                summarySubTotalDec.ToString("c") + "\r\n" + "\r\n" +
                "The culmulative Sales Tax is " +
summarySalesTaxDec.ToString("c") + "\r\n" + "\r\n" +
                "The culmulative Total is " + summaryTotalDec.ToString("c");
        }
```

```
        private void okButton_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```