CISP 41

Programming in C#

Good job!

Project Evaluation Sheet

Student Name: ___________ Bryant Tunbutr ____________________ Project Number: ______1__________

Project Name: _______BtunbutrProject1_____ Visual Studio Version:___2008________

Date Due: _______9/20/2012______________ Date Turned In: ______9/20/2012____

Above to be completed by student

Points (40 Possible)

Correctness/Efficiency:

Output is accurate ____________

Meets all requirements ____________

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: ______+3______

Time liness: ____________

Project Score: 43/40

CISP 41 Programming in C#

Project specification

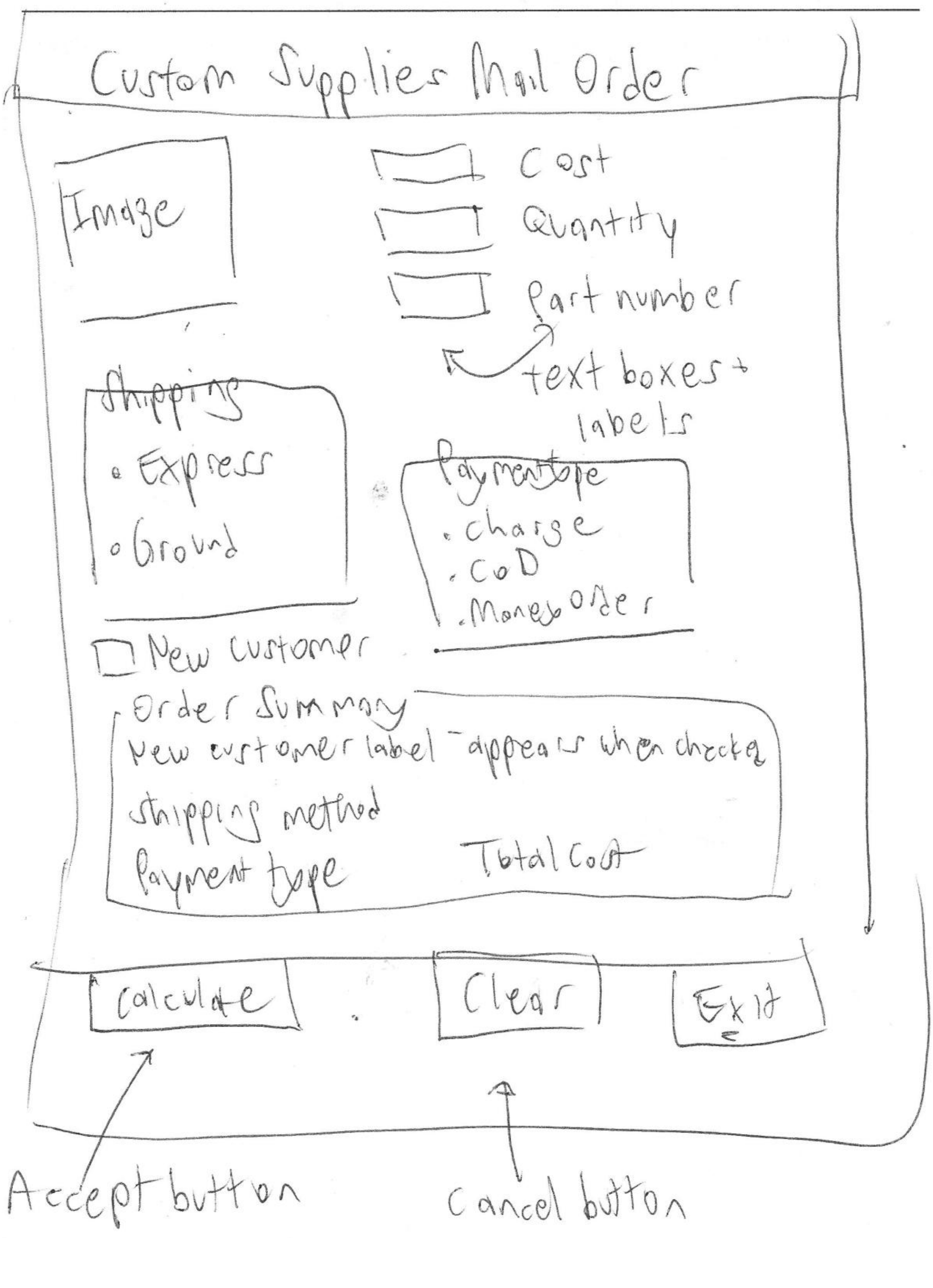
This software is intended to summarize, calculate, and display calculate shipping costs for Custom Supplies Mail Order. It is designed to be run in Visual Studio 2008 using the C# coding. It uses user input including the method of shipping and payment, quantity, and part number. It displays total cost, method of payment and shipping, and the part number. It is also designed to catch order errors and acknowledge a new customer.

CISP 41 Programming in C#

Project status

The project is completed and finalized

Sketch of user interface



CISP 41 Programming in C#

Objects and Properties Plan for ____BTunbutrProject1_____Form

Object	Property	Setting
exitButton	**Name Text**	**E&xit**
clearButton	**Name Text**	**C&lear**
calculateButton	**Name Text**	**calculate**
groupBox1	**Text**	
BTunbutrProject1	**Name Text**	*Form*
label1	**Name Text**	**Sales**
label6	**Text**	**Amount Due**
totalAmountTextBox	**Name Text**	**totalAmountTextBox**
label5	**Text**	**Shipping**
Radiobutton	**Text**	**COD, Check, Money order**
Check box	**Text**	**New customer**
Pic	**Image**	**Stretch**

CISP 41 Programming in C#

Event Plan for ____ BTunbutrProject1_______Form

Object	Event	Action - Pseudocode
calculateButton	Click	Declare the variables. try Convert the input Quantity to numeric. try Convert the input Price to numeric. Calculate Extended Price =Quantity * Price Format and display sale output. Format and display summary values. catch any Quantity exception Display error message and reset the focus to Quantity. catch any Price exception Display error message and reset the focus to Price. Display shipping and payment method
clearButton	Click	Clear each text box except Summary fields. Set the focus in the first text box.
exitButton	Click	Exit the project.

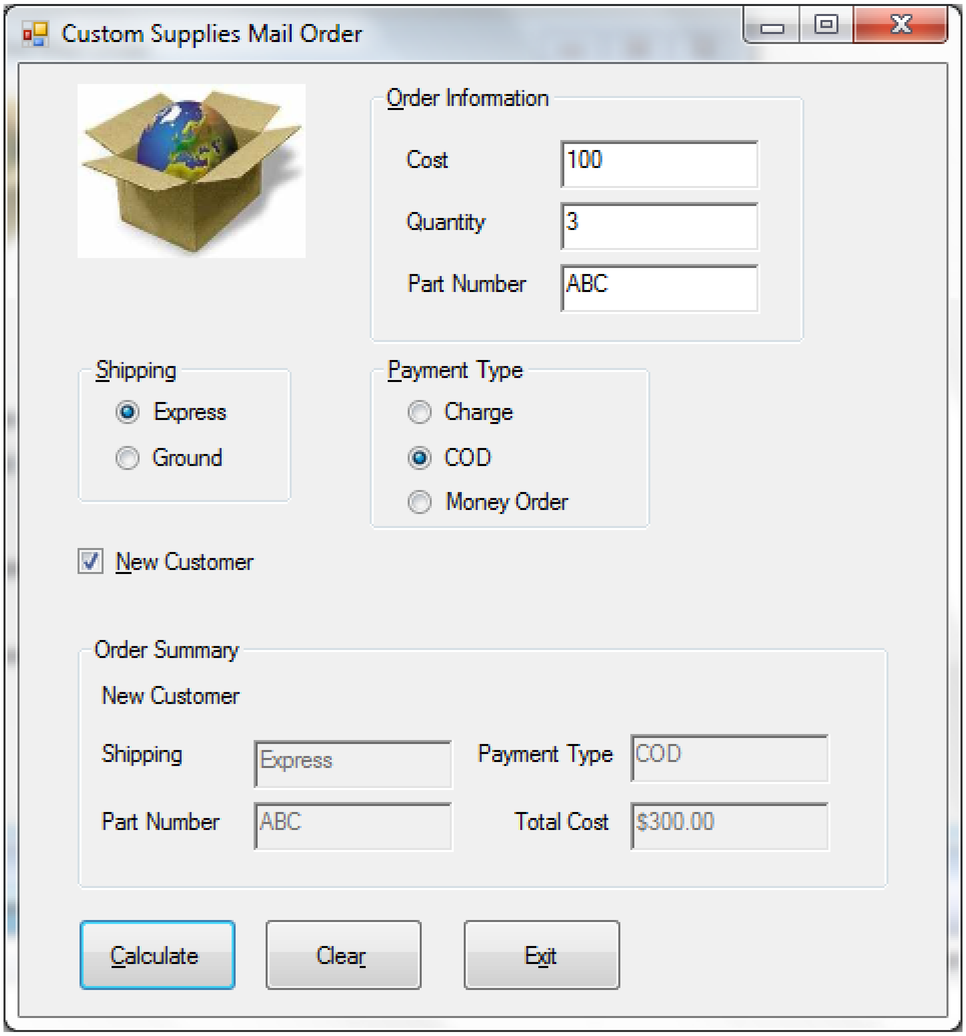
CISP 41 Programming in C#

Test cases and captured screens

Test case #1

```
Should mention radio buttons and check box
```

Cost = 100
Quantity = 3
Part Number = ABC



Custom Supplies Mail Order

Order Information

Cost: 100
Quantity: 3
Part Number: ABC

Shipping
- ● Express
- ○ Ground

Payment Type
- ○ Charge
- ● COD
- ○ Money Order

☑ New Customer

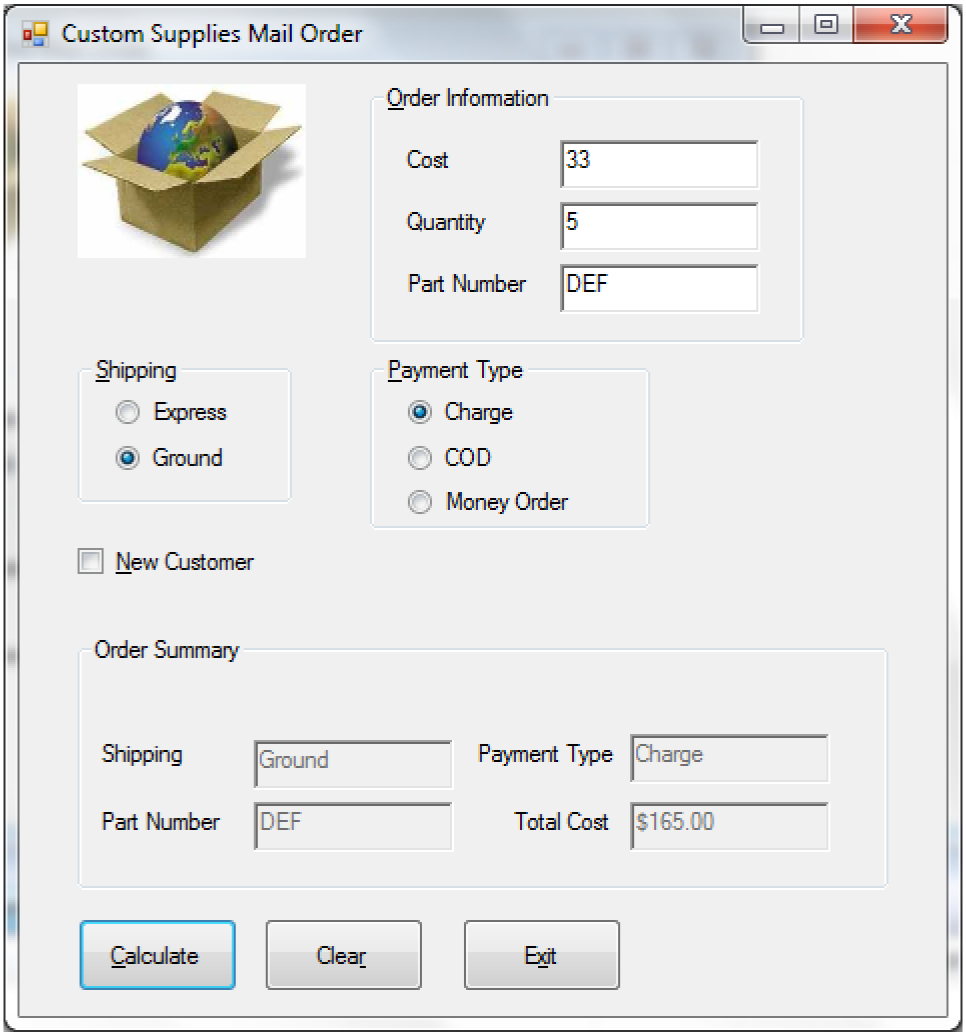
Order Summary

New Customer

Shipping: Express
Payment Type: COD
Part Number: ABC
Total Cost: $300.00

Test case #2

Cost = 33
Quantity = 5
Part Number = DEF



Order Information

Cost 33
Quantity 5
Part Number DEF

Shipping
○ Express
● Ground

Payment Type
● Charge
○ COD
○ Money Order

□ New Customer

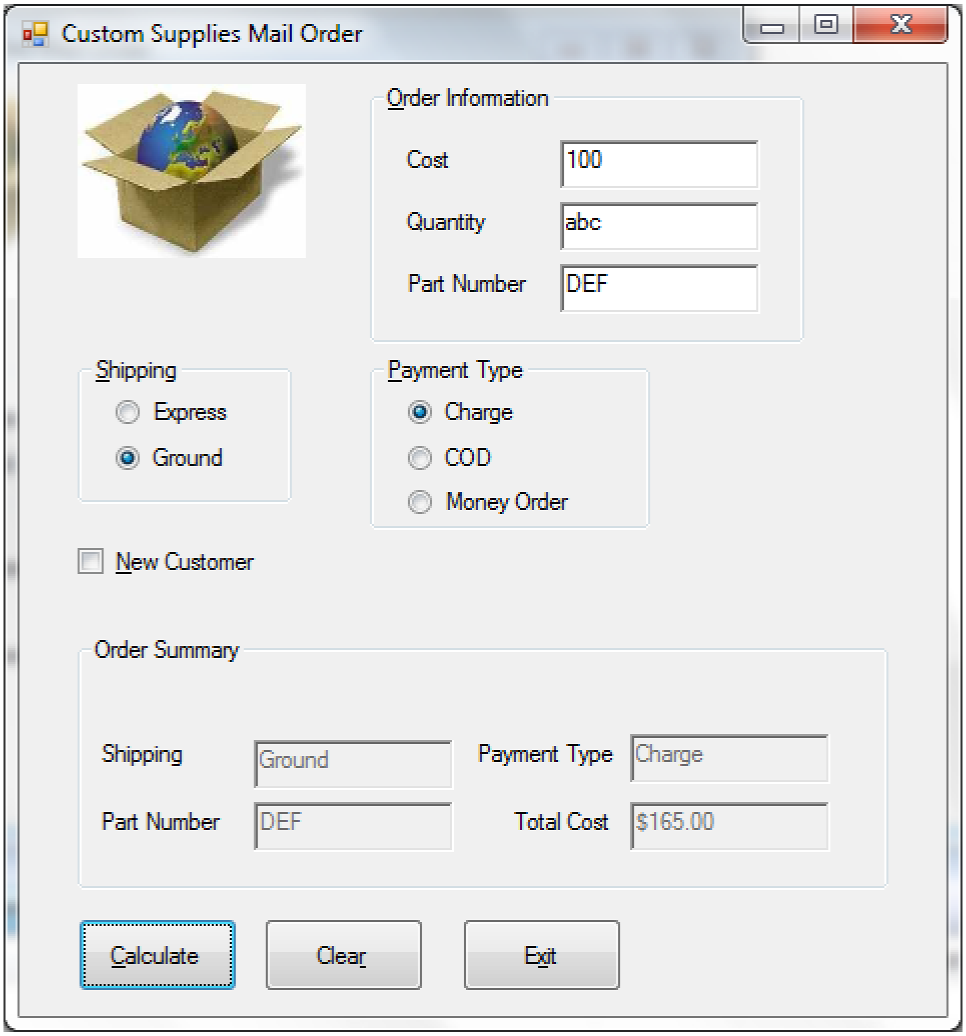
Order Summary

Shipping Ground
Payment Type Charge
Part Number DEF
Total Cost $165.00

Center the three buttons.

Test case #3

Cost = 100
Quantity = abc
Part Number = DEF



Custom Supplies Mail Order

Order Information

Cost	100
Quantity	abc
Part Number	DEF

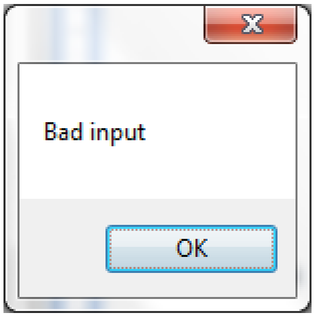
Shipping
○ Express
● Ground

Payment Type
● Charge
○ COD
○ Money Order

□ New Customer

Order Summary

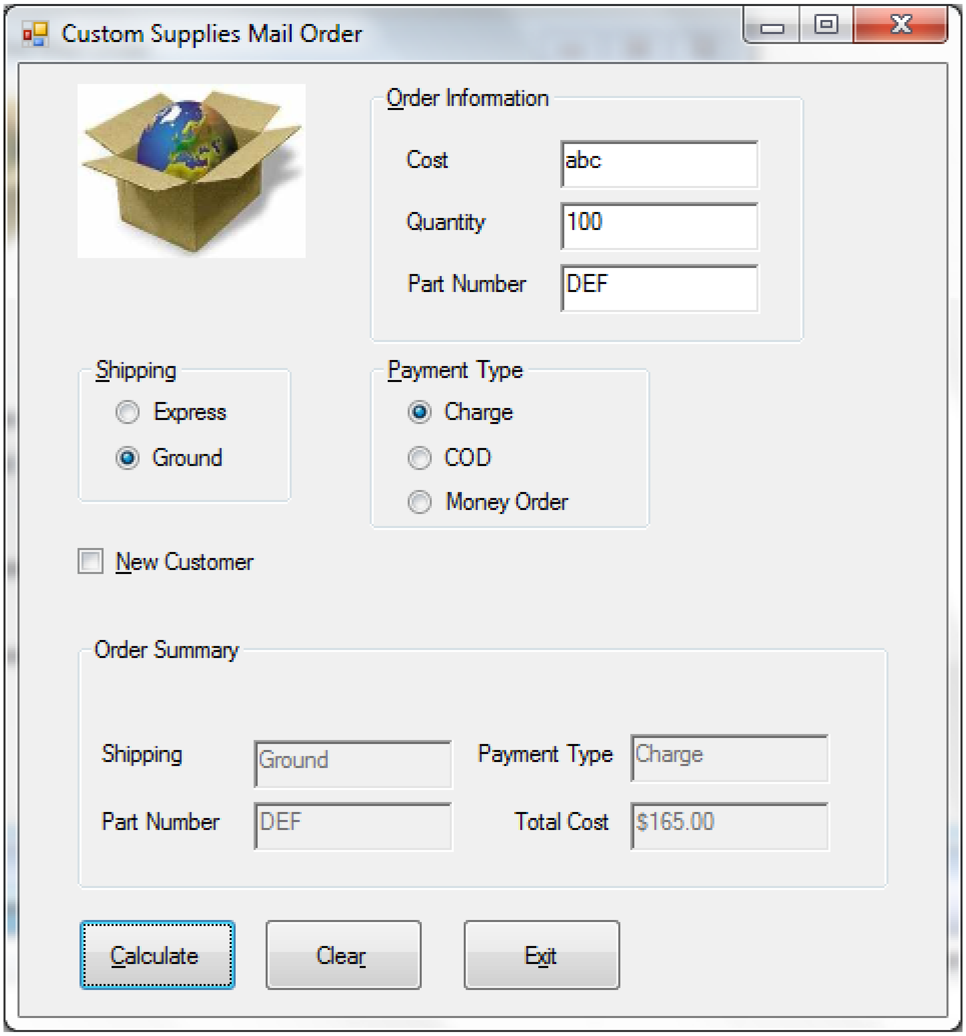
Shipping	Ground	Payment Type	Charge
Part Number	DEF	Total Cost	$165.00



Bad input

Test case #4

Cost = abc
Quantity = 100
Part Number = DEF



Custom Supplies Mail Order

Order Information

Cost	abc
Quantity	100
Part Number	DEF

Shipping

○ Express
● Ground

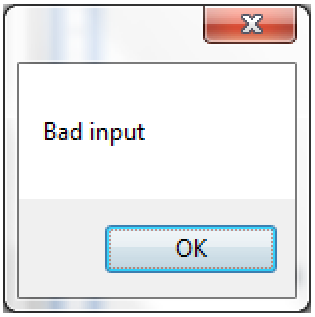
Payment Type

● Charge
○ COD
○ Money Order

□ New Customer

Order Summary

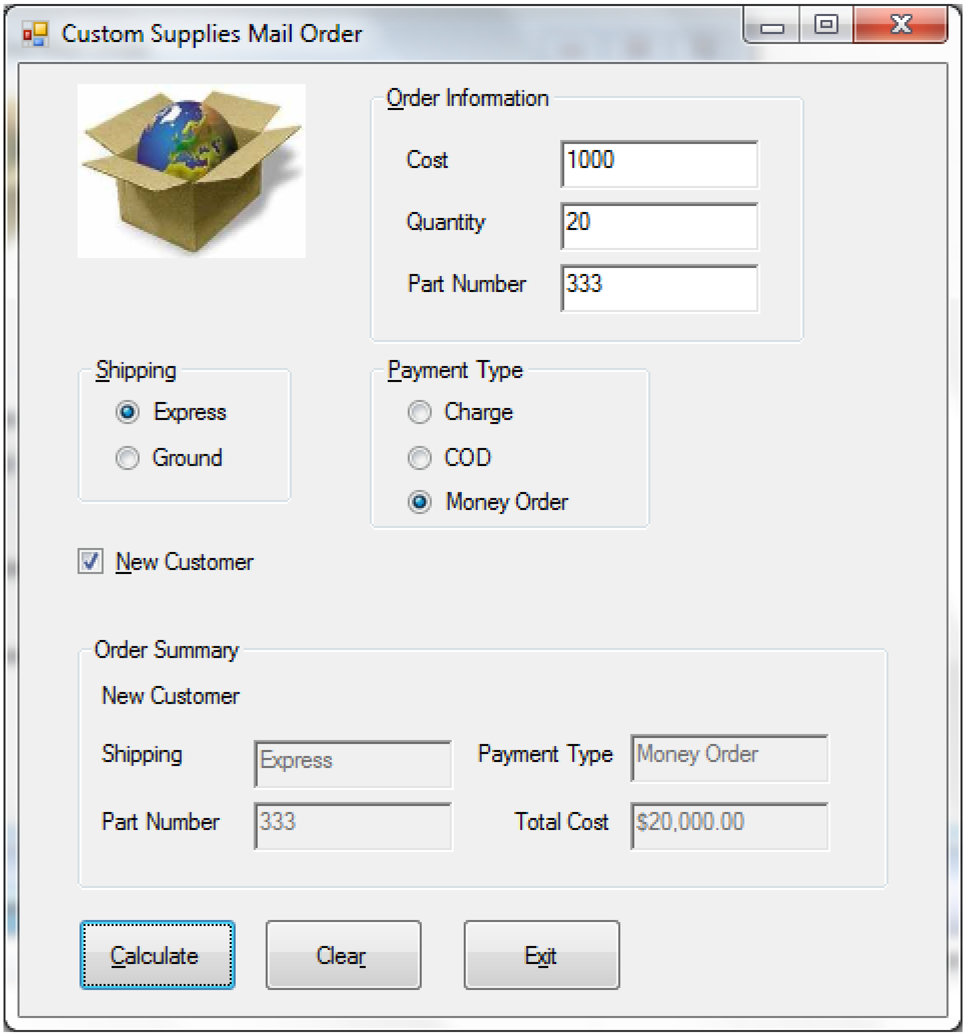
Shipping	Ground	Payment Type	Charge
Part Number	DEF	Total Cost	$165.00



Bad input

Test case #5

Cost = 1000
Quantity = 20
Part Number = 333



Order Information

Cost 1000

Quantity 20

Part Number 333

Shipping

● Express

○ Ground

Payment Type

○ Charge

○ COD

● Money Order

☑ New Customer

Order Summary

New Customer

Shipping Express

Payment Type Money Order

Part Number 333

Total Cost $20,000.00

CISP 41 Programming in C#

Source code

```
/*
* Project: BTunbutrProject1
* Programmer: Bryant Tunbutr
* Date: September 19 2012
* Description: This project summarizes, calculates, and displays the charges for shipping
a package.
* 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 WindowsFormsApplication7
{
    public partial class BTunbutrProject1 : Form
    {
        public BTunbutrProject1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void costLabel_Click(object sender, EventArgs e)
        {

        }

        private void quantityLabel_Click(object sender, EventArgs e)
        {

        }

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

        // Select the shipping options by using radio buttons.
        private void expressRadioButton_CheckedChanged(object sender, EventArgs e)
        {
            shippingTextBox.Text = "Express";
        }

        private void groundRadioButton_CheckedChanged(object sender, EventArgs e)
        {
            shippingTextBox.Text = "Ground";
        }

        // Select the payment options by using radio buttons.
        private void chargeRadioButton_CheckedChanged(object sender, EventArgs e)
        {
            paymentTextBox.Text = "Charge";
        }

        private void codRadioButton_CheckedChanged(object sender, EventArgs e)
        {
            paymentTextBox.Text = "COD";
        }

        private void moneyOrderRadioButton_CheckedChanged(object sender, EventArgs e)
        {
            paymentTextBox.Text = "Money Order";
        }

        private void partNumberTextBox_TextChanged(object sender, EventArgs e)
        {

        }

        /* Select if new customer. This label only appears in
        the order summary if the checkbox is clicked.
         */
        private void newCustomerLabel_Click(object sender, EventArgs e)
        {
            newCustomerLabel.Visible = newCustomerCheckBox.Checked;
        }

        // Clicking the calculate button does calculations and displays the order summary.
        private void calculateButton_Click(object sender, EventArgs e)
        {
            // Display the part number in the order summary.
            partTextBox.Text = partNumberTextBox.Text;

            // Calculate the total amount due.
            // Declare the variables.
            int quantityInt;
            decimal priceDec, extPriceDec;
            try
            {
                // Convert input values to numeric and assign to variables.
                quantityInt = int.Parse(quantityTextBox.Text);
                priceDec = decimal.Parse(costTextBox.Text);
                // Calculate values.
                extPriceDec = quantityInt * priceDec;

                //Display the results
                totalCostTextBox.Text = extPriceDec.ToString("C");

            }
            // To catch bad input.
            catch
            {
                MessageBox.Show("Bad input");
            }
        }

        private void clearButton_Click(object sender, EventArgs e)
        {
            // Clear the text boxes.
            costTextBox.Text = "";
            quantityTextBox.Text = "";
            partNumberTextBox.Text = "";
            shippingTextBox.Text = "";
            paymentTextBox.Text = "";
            partTextBox.Text = "";
            totalCostTextBox.Text = "";

            costTextBox.Focus();

        }

    }
}
```