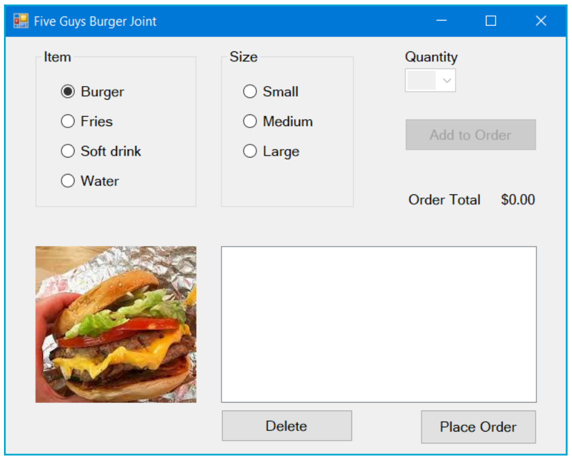
Assignment # 2 (20 points) DUE: Specified on Canvas

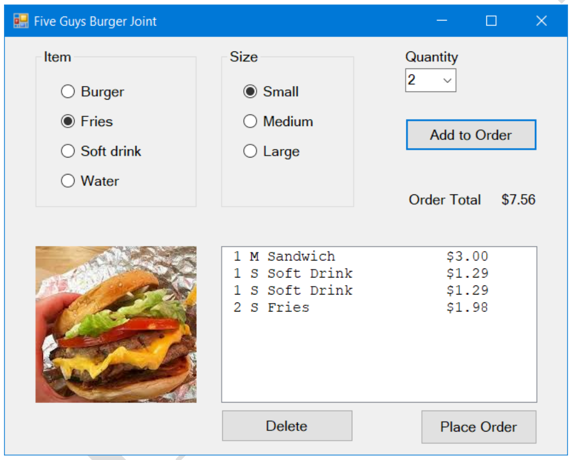
Assignment Purpose

This assignment aims to create a C# class representing items ordered in a fast-food restaurant and to calculate the total cost of the items ordered. You will practice using C# and GUI interface design and implementation. You may work with one other student on this assignment.

Mandatory Instructions

Begin by creating a C# project on the ZIP disk, USB flash drive, or your computer's HD. The project's name should be one or two last names followed by Lab2 (e.g., Smith-Assignment2 or Smith-Jones-Assignment2). You will first want to build the form as indicated on the left below. The form on the right shows how it will appear after a few items have been ordered.





```
1 M Sandwich          $3.00
1 S Soft Drink        $1.29
1 S Soft Drink        $1.29
2 S Fries             $1.98
```

Note the following requirements for this form:

1. Every control should use a 10-point font. The easiest way to accomplish this is to set the form's font to be 10-point. That will then serve as the default for other controls on the form.
2. The Name property for controls (other than Form1) should use our standard naming convention (e.g., grpItem, radSandwich, btnOrder, lblTotal, lblTotalLabel, cboQuantity, etc.).
3. None of the radio buttons should be selected initially.
4. The following controls should initially be disabled: the radio buttons for Size, the combo box for Quantity, both buttons.
5. For the quantity combo box, set the DropDownStyle property to be DropDownList. Then click on the Items property and then the three dots at the end (ellipsis). Enter in the integers from 1 to 10 on separate lines.
6. The total shown is two labels. One is a static label that won't be changed ("Total"). The other label will be updated during runtime, so it should be given a meaningful name (e.g., lblTotal). The TextAlign property for both should be set to TopRight.
7. The list box should use Courier New as the font.
8. Place Order button should close the form.
9. Add ImageBox and import the given image file as a resource.

Try running the program, and you should see that the Item radio buttons will allow only one to be selected. That's because these buttons are located in the same group box.

Declare two private int variables called formItem and formSize in the Form class. These will be used to represent the currently selected item and Size. You'll also need a private decimal item totalCharge.

Add a new class called PurchasedItem to the project. This class should have private integer data representing the item (1 through 4), Size (1 through 3), and Quantity (1 through 10). There is also a static string value called format and a static 2-dimensional decimal array that contains the per-item cost values (small/med/large for each of the items). Here is how to declare and initialize the static data using a static constructor:

```
    private static decimal[,] itemCost; // 2-d array of item costs
    private static string format;

    static PurchasedItem()
    {
        format = "{0, 2}  {1, 1} {2, -15} {3, 10:C}";
        itemCost = new decimal[4, 3]
        {
            {2.50M, 3.00M, 3.50M},
            {0.99M, 1.29M, 1.49M},
            {1.29M, 1.40M, 1.60M},
            {0.00M, 0.00M, 0.00M}
        };
    }
```

You should define a public read-only decimal property called cost that returns the total cost for this item (quantity times unit cost). Override the ToString() function to return a string such as the following:

```
 2  M Fries                 $2.58
```

You can use the String.Format function to format this string using the private variable *format*.

When the program runs, this is what should happen:

1. If the Water button is clicked, enable the Small button, select it, disable the other Size buttons, and enable the quantity combo box and the Order button. Set formItem to 3 and formSize to 0. Select the "1" in the combo box by executing the following statement:
   ```
   this.cboQuantity.SelectedIndex = 0;
   ```

2. Write a private function with no arguments called enableSizes. This function should enable all size buttons but should not select any.
3. If the Sandwich button is clicked, call enableSized to enable all buttons but don't select any. Disable the quantity combo box and the order button. Set formItem equal to zero. Do something similar for the Fries and Soft Drink buttons (where formItem = 1 for Fries and 2 for Soft Drink).
4. If any size button is pressed, enable the quantity combo box and the Order button. Select the "1" in the combo box. Set the value of formSize appropriately (0 = Small, 1 = Medium, 2 = Large).
5. If a line in the list box is selected, enable the Delete button.
6. If the Order button is clicked, create a new PurchasedItem object and add it to the list box. Update the total charge and display it on the form as a currency value (using ToString("C")).
7. If the Delete button is clicked, subtract the selected item's cost from the total charge and update the form. Remove the selected item from the list box.

When you have fully debugged your code, copy the compressed project folder to Canvas. Each group member should upload the solution, and each will supply their grade sheet for this project.

Code Documentation

Each source file should contain a documentation header with a description, author name, and class information (CS3160, Fall/Spring 20XX). Each function should have a documentation header with a minimally function name, description, parameters, return value. Use proper program style at all times, which means indentation, alignment, and whitespace. Utilize self-documenting code which means use mnemonic, i.e., meaningful variable/function/namespace/class names, etc.

What to turn in?

Submit a compressed (.zip) solution folder via Canvas.
Make sure that all necessary files to compile/link/execute your projects are provided in your solution folder.
Here are some files that may be required:

1. All required C# source files (.cs)
2. makefile (if the solution requires it)
3. The entire Visual Studio 2019 solution folder (if the solution requires it)