LAB ASSIGNMENT #4

Due Date: **Week 10** **Marks/Weightage: 40/10%**

Purpose: The purpose of this Lab Assignment is to:

- Practice the use of various GUI controls, properties and event handlers.

References: Read the course's text book **chapter 12 and 13 – GUI and Event Handling** and the lecture notes/ppts. This material provides the necessary information that you need to complete the exercises.

Instructions: Be sure to read the following general instructions carefully:

This lab should be completed individually by all the students. You will have to demonstrate your solution in a scheduled lab session and submitting the project **through drop box link on e-Centennial**.

You must name your solution according to the following rule:
FirstName-LastName_SectionNumber_COMP123_Labnumber
For Example: **Joh-Smith_Sec001_COMP123_Lab01**

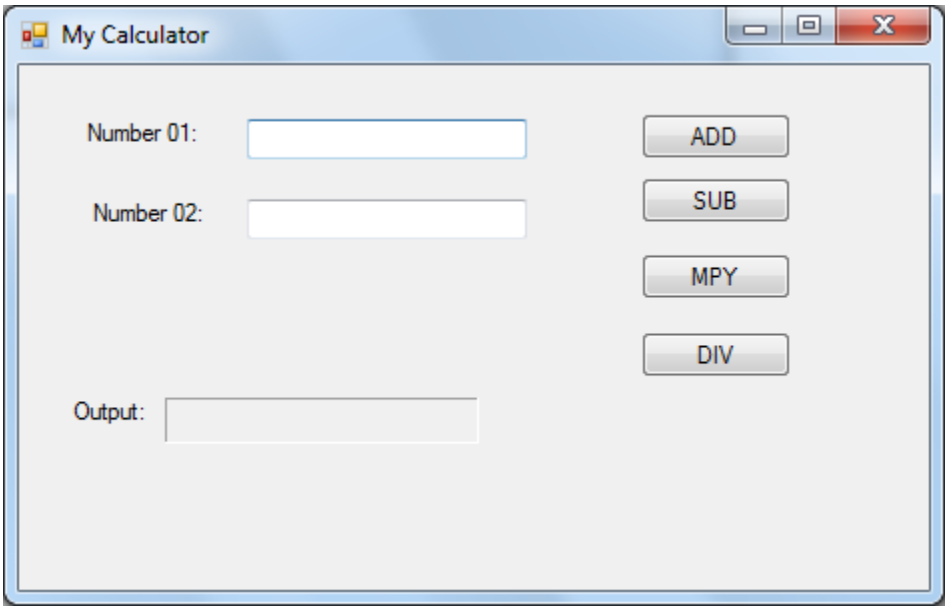
Each exercise should be placed in a separate namespace named firstname-last-name_*exercise1*, firstname-last-name_*exercise2* etc.

Submit your assignment in a **zip file** that is named according to the following rule:
FirstName-LastName_SectionNumber_COMP123_Labnumber.zip
Example: **Joh-Smith_Sec001_COMP123_Lab01.zip** *(if your section is 001..)*

Apply the naming conventions for variables, methods, classes, and packages:
- *variable names* start with a *lowercase* character for the first word and uppercase for every other word
- *classes* start with an *uppercase* character of every word
- namespaces use only *lowercase* characters
- *methods* start with a *uppercase* character for the first word and uppercase for every other word

Exercise #1: *[15 marks]*

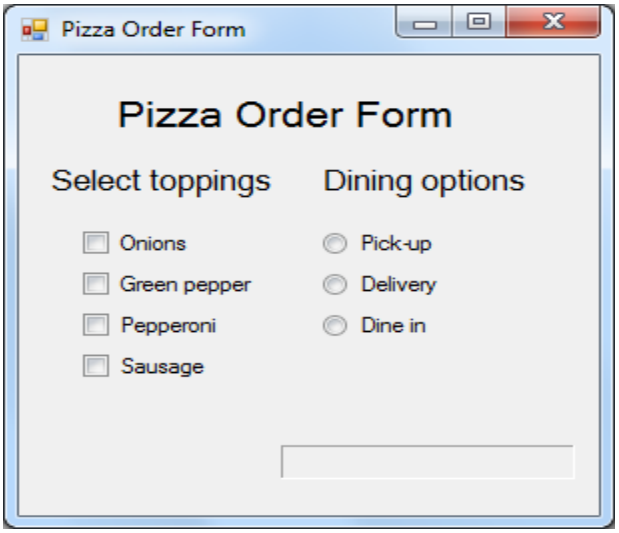
Following Calculator Window form application has been covered in the class. Solution is posted onto e-centennial. During the lab, we only completed the **Add** button functionality.



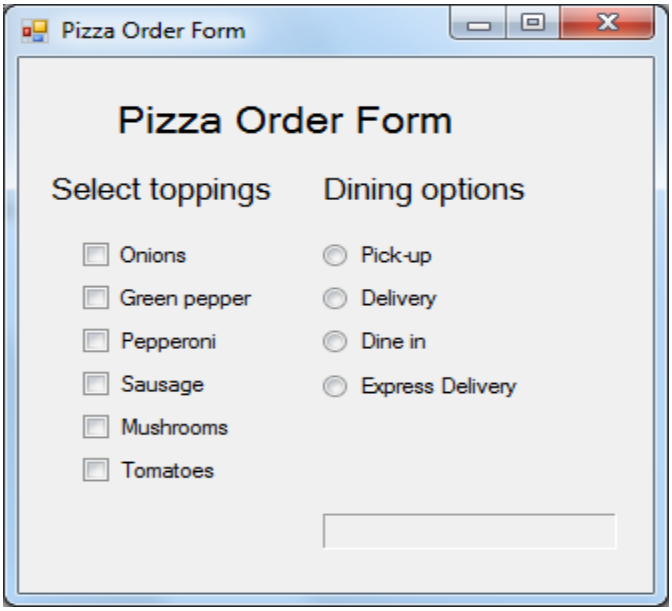
You need to provide the functionality for the remaining buttons – **SUB** for subtracting number 02 from number 01, **MPY** for multiplying number01 and number02, **DIV** for diving number01 by number02. Also you need to add exception handling here.

Exercise #2:

Following Pizza Order Window form application has been covered in the class. Solution is posted onto e-centennial.



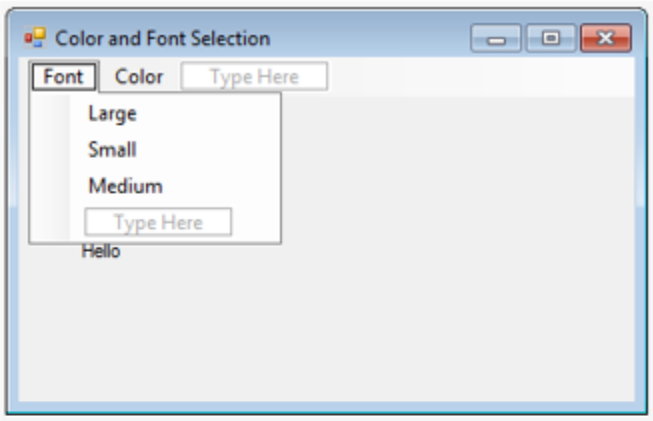
You need to extend the above application by adding the following controls as shown in the screen shot below.

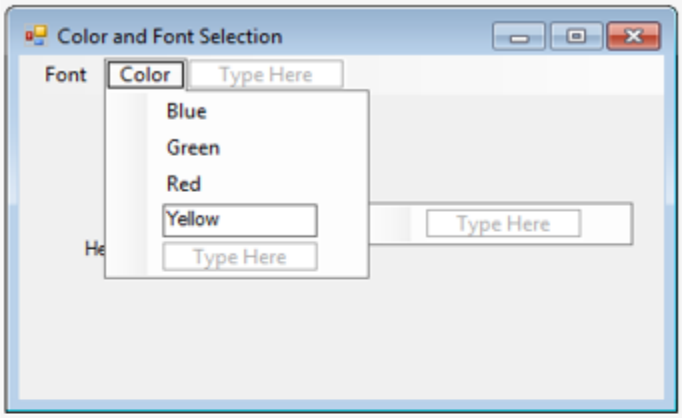


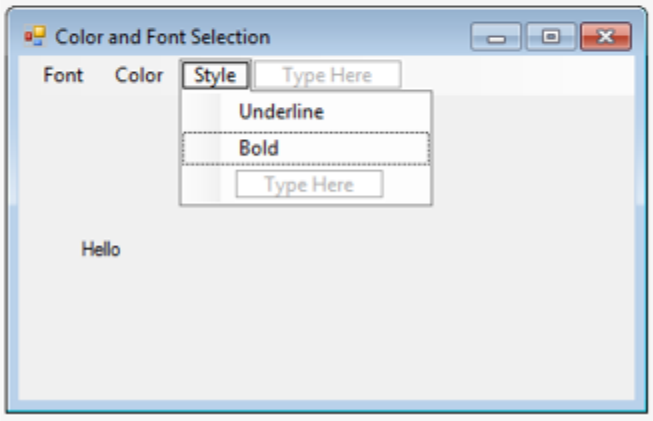
User selects the toppings from the menu and also choose the dining options, then he will be charged accordingly.
Price calculation requirements:
-Base price for pizza is 10.00 dollars
-Every veg topping costs dollar each
-Every non-veg topping costs two dollars.
- Charges for the express delivery is $10.00

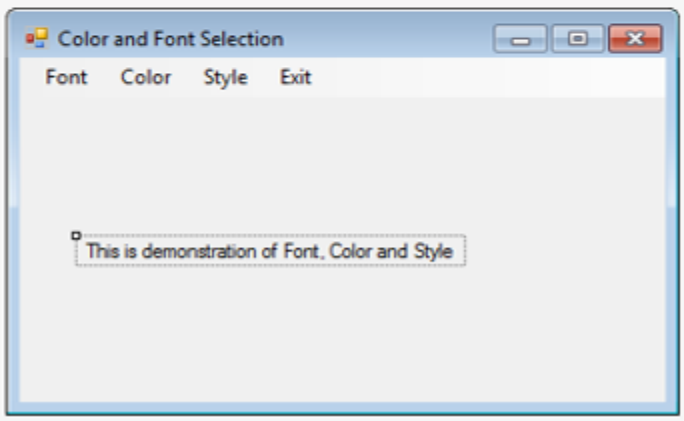
Exercise #3:

Refer the code example solution folder – **MenuStripDemo** (in chapter 12 Code example), you need to extend that application as shown below. Add extra menu trip items shown below in the screenshots. Add corresponding event handler's for functionality. Replace the label text – **Hello** with **This is demonstration of Font, Color and Style.** When user click a menus strip item, the text should change its font, color and style accordingly. When user click Exit, it should exit the application.



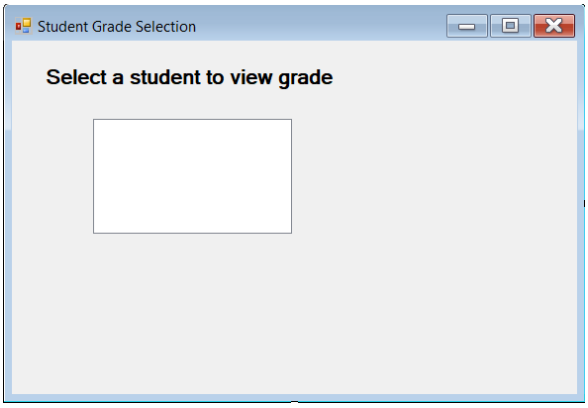


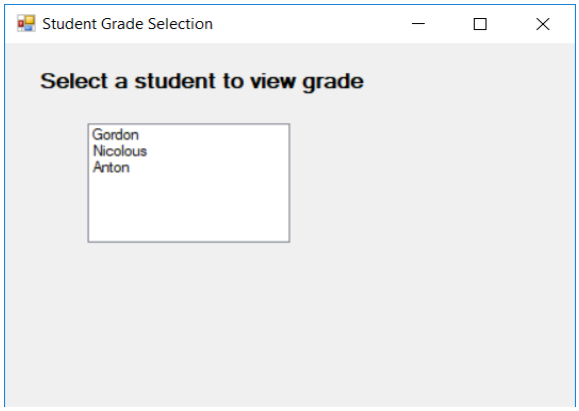


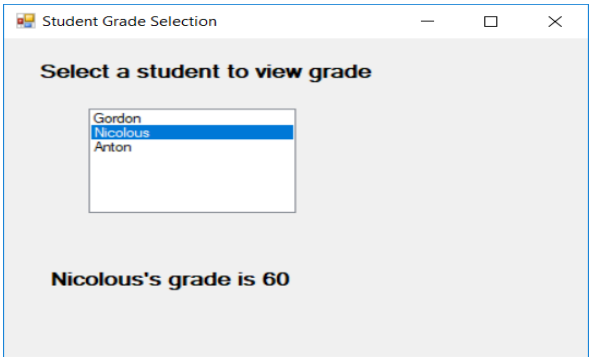


Exercise #4:

Refer the code example solution folder – **AddRangeObjectDemo** (in chapter 12 Code example), below application is similar to the above.
Build your GUI as shown below. User clicks a student and grades of that student should be displayed.
You need to define **Student** class in your application which is having Name and Grade properties. Define a constructor in the class and override ToString() method. When application is started, listbox is populated with the name of the students. When user clicks a student name and grades of that student should be displayed.







Evaluation:

Functionality	
Correct implementation of classes (instance variable declarations, constructors, getter and setter methods etc.)	70%
Correct implementation of driver classes (declaring and creating objects, calling their methods, interacting with user, displaying results)	20%

Comments, correct naming of variables, methods, classes, etc.	5%
Friendly input/output	5%
Total	100%