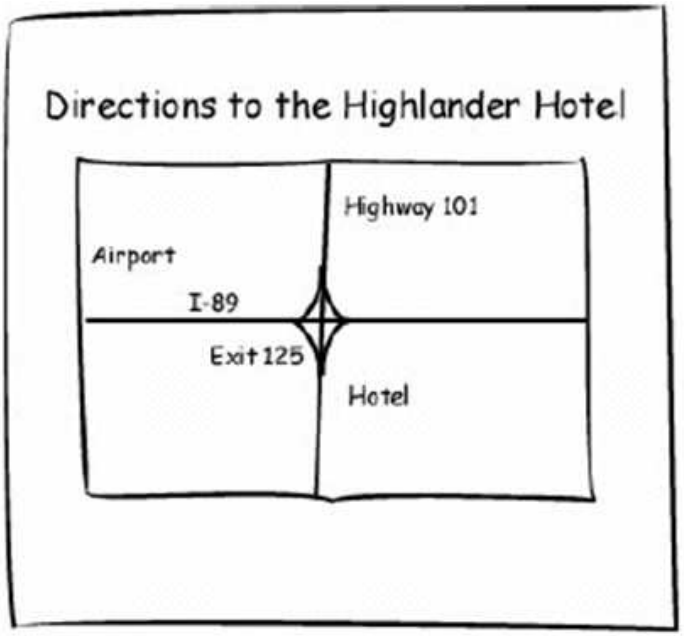
Lecture 3 In-class Lab

Name:_____

Part A: Creating an application in Visual Basic: You and your lab partner will be working through the tutorials in chapter 2. In part A, you will be working tutorials 2-1 through 2-8 in the textbook. You can refer to the textbook for detail directions, but you might be able to get by with the following outline. In part A, your task is to write a simple application to display a map to the Highlander Hotel (see the figure).



1) Create a New Project

2) Add the following controls from the Toolbox, and modify their properties as:

Form
- Name: Form1
- Text: "Directions"

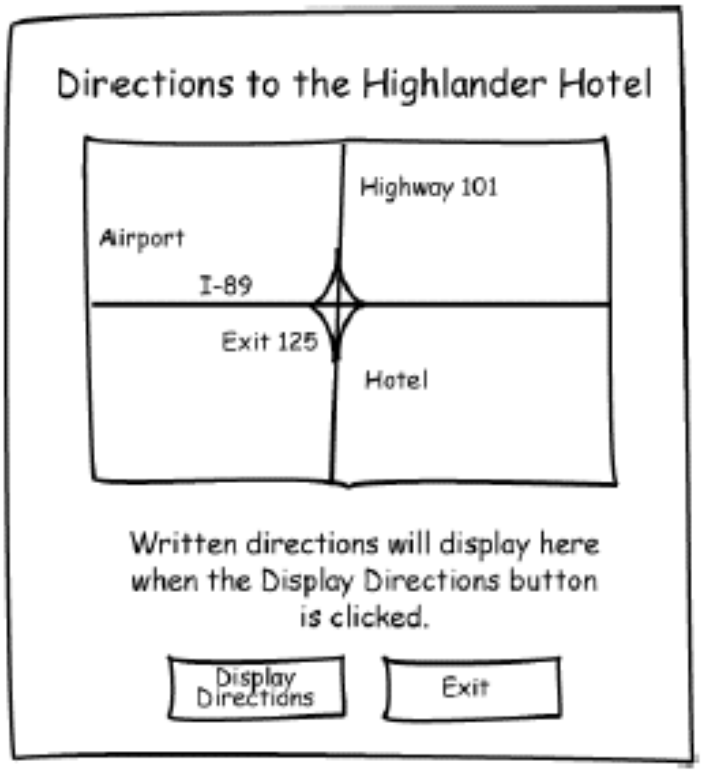
Label
- Name: Label1
- Text: "Directions to the Highlander Hotel"
- TextAlign: MiddleCenter
- Font: Microsoft sans serif, bold, 18 point

PictureBox
- Name: PictureBox1
- Image: Import the Project resource file HotelMap.jpg from the P:\Math-CS\810-030\common directory
- SizeMode: StretchImage

3) Save All your project on the P:\Math-CS\810-030\<your username> .

4) Run the application by using the Debug | Start Debugging

Part B: Augment the Hotel Application with a couple buttons to:
- **view written directions when the "Display Directions" button is pressed**
- **exit the application**



1) Add the following controls from the Toolbox, and modify their properties as:

Label
- Name: lblDirections
- Text: "Traveling on I-89, take Exit 125 onto Highway 101 South. The hotel is on the left, just past the I-89 intersection. Traveling on Highway 101 North, the hotel is on the right, just before the I-89 intersection."
- AutoSize: False
- Visible: False (Run the application to see that the label does not appear since it is "invisible")

Button
- Name: btnDisplayDirections
- Text: "Display Directions"

Button
- Name: btnExit
- Text: "Exit"

2) When the btnDisplayDirections button is clicked, we want the lblDirections label to become visible. Therefore, we must add an event procedure to handle the user event of clicking on the btnDisplayDirections button. The following steps will allow you to add code to handle this event:

- Double-click on the btnDisplayDirections button on the form to automatically generate a template for the subroutine for this event handler as part of Form1's code. It will look something like:

```
Private Sub btnDisplayDirections_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnDisplayDirections.Click

End Sub
```

- Add code in the body of the subroutine (on the line before the "End Sub") to set the label lblDirections Visible property to True. This can be do by adding the following assignment statement:
  lblDirections.Visible = True

Note: As you type the line of code, Visual Basic shows you possible options at each point to assist you. The resulting event handler code will look something like:

```
Private Sub btnDisplayDirections_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnDisplayDirections.Click
        REM Make the directions visible
        lblDirections.Visible = True
End Sub
```

- Run the application (Debug | Start Debugging) to verify that the event is handled correctly.

3) When the btnExit button is clicked, we want the application, Form1 to close. Therefore, we must add an event procedure to handle the user event of clicking on the btnExit button. The following steps will allow you to add code to handle this event:

- Double-click on the btnExit button on the form to automatically generate a template for the subroutine for this event handler as part of Form1's code. At this point the complete code will look something like:

```
Public Class Form1

    Private Sub btnDisplayDirections_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnDisplayDirections.Click
            REM Make the directions visible
            lblDirections.Visible = True
    End Sub

    Private Sub btnExit_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnExit.Click


    End Sub
End Class
```

- Add code in the body of the subroutine that Handles btnExit.Click to close the form, Form1. This is done by adding the line: Me.Close( )

Note: Since we are inside of the code for Form1, "Me" refers to Form1 which is the application.

- Run the application (Debug | Start Debugging) to verify that the event is handled correctly.

4) Save All, the application and exit Visual Studio

Each team on a laptop needs to turn in a piece of paper with the following: Names of the two partners on a laptop, and P: subdirectory where the application is located.

Both partners might also want to also store the application on USB flash drives, too.