**In-Class Worksheet #3**

## Creating a Windows Forms application

You should have Visual Studio 2008 open.

1. Create a new Project either from the shortcut on the Start page or through the file menu,
2. Select the type to be a C# Windows Forms Application.
3. In the Location textbox, enter your desired hard drive location. Note that this will be on the mounted z: drive in the CSE environment. Name the Project *Lastname*Fav5Web where *Lastname* is replaced with your name or an alias.
4. Select OK.
5. Open Solution Explorer (View->Solution Explorer or the shortcut Ctrl-Alt-L). You will see tree views for the Properties, References and Form1.cs. Expand the Form1.cs branch. It includes a Form1.Designer.cs entry. This is an additional file in your project (you can verify this using Windows Explorer to look at the files created). This file is a normal C# source code file that is generated and updated automatically by Visual Studio as you work with the Visual Designer.
6. A Form1.cs file is also created and opened automatically in the Visual Designer.  Right-click (almost) anywhere in this window and select View Code. This opens the file Form1.cs in the regular source code editor format. Note that the Designer View is primarily displaying the Form1.Designer.cs file.
7. In the source code view for Form1.cs you will notice that a Class called Form1 is defined. Form1 has a base class called Form (actually it is System.Windows.Forms.Form). It also has a new keyword called **partial** in its definition. This indicates to the compiler that one part of the class is defined in this file (currently only the constructor).
8. Go back to the Solution Explorer and right-click on the file Form1.Designer.cs and select View Code. This opens another source code view, but this time with the source for Form1.Designer.cs. Note that it also defines a Class (a type) named Form1 in the same namespace. Fortunately it is also marked as **partial**, so it is the rest of the definition of Form1 from the Form1.cs file. It has added a member variable (components) and two methods (Dispose and InitializeComponent) to the class definition of Form1. If you go back to the source code view for Form1.cs you will see that the first thing the constructor does is call the InitializeComponent method.
9. Close the Form1.Designer.cs file. You should avoid editing this file directly. Unfortunately, they placed the Dispose method in this file, so you may need to edit it occasionally.
10. The proper way to think of these two files is that the compiler will merge the two sources together and then compile them. In general you can have several files defining a class as long as they are all available during compilation.

## Create MyFavFiveWeb application:

We are going to create a simple web browser with a constrained set of predetermined web sites to visit as illustrated to the right. It has a large pane to view the web page with a combo box and button below to select the web page to display.

11. Right-click on Form1.cs in the Solution Explorer and select Rename. Change Form1.cs to MyBrowser.cs. You will be prompted to change the name throughout the solution. Click yes.
12. In the Visual Designer "MyBrowser.cs [Design]" open the Toolbox (View->Toolbox). Note the title bar of this panel like many others has an options drop-down combo box, a little icon of a push-pin and a close icon. The push-pin is the Auto-Hide
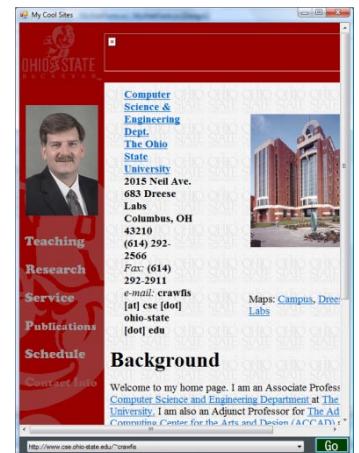
feature. If upright, the panel will stay and occupy space. If clicked to be horizontal, the toolbox will iconify over to the side and open when the mouse is moved over it. Experiment with the various options of the title bar.

13. With the Visual Designer in Visual Studio you can develop your user interface or code using a more ***declarative programming*** style. Select the title bar of Form1 in the designer. Even though you changed the type or class name, Form1 still is the name of the particular instance of MyBrowser. Bring up the **Properties** panel (View->Properties Window). This panel has a combo box at the top where you can select the component, followed by a toolbar and then a list/tree view of all of the properties that can be set for this component. Since MyBrowser is the only component right now that is all the combo box has. The toolbar allows you to display the properties grouped into categories or alphabetically (A->Z). It also allows you to look at Properties or Events. Properties are just that, publically exposed properties in a control (your MyBrowser is not registered as a control, so it only has the Form properties). We will look at events later in the course.

14. Scroll down the list of properties and change ***Text*** to something more meaningful. This is the text displayed on the title bar of your application.

15. Change the ***Size*** property to 800 by 1024.

16. Other properties you might be interested in are BackColor, StartPosition and WindowState.

17. From the Toolbox, expand the Containers tree and select the SplitContainer control and drag it onto your form.

18. Set the SplitContainer property **Orientation** to Horizontal. Note, three new controls were added to the form, the splitcontainer and its two panels.

19. Most controls have a **(Name)** property and a **GenerateMember** property. These indicate whether a member variable should be created in your class and if so what the variable name should be. For this control, we will not be doing anything with it at run-time, so set **GenerateMember** to false. Note that there is also a **Modifier** property to indicate whether it should be a private, protected, etc. accessible member variable.

20. Drag the splitter (the small line in the split container) to be about 30 pixels or so from the bottom.

21. From the Common Controls of the Toolbox drag a WebBrowser control to the upper panel.

22. Drag a Button to the bottom panel.

23. Rename the button **(Name)** property to *goButton*. Set the **Text** to *Go.*

24. Set the **Dock** Property to the *Right*. The Dock and Anchor properties dictate how the control will react when its parent control is resized. The split containing is docked to fill the entire interior of the form. The WebBrowser is docked to Fill the entire upper panel.

25. Drag a ComboBox to the bottom panel.

26. Set the Dock Property to Fill. I should point out that things are a little finicky and order dependent in this. If add the ComboBox and then the Button you will get different behaviors.

27. On the Items Property (of the ComboBox), click the little … button. This will bring up a text editor. Type in your five favorite web sites in this box, each one on a new line.

28. Select the splitContainer and in its Panel2 properties, select a Padding of 5,5,5,5. Note the change. Adjust the splitter bar to your liking.

29. Build and Run your application. You will note that the combo box items can be selected. The Go button does nothing, and no web page is displayed. We now have a working GUI, but no logic.

30. Double-click the Go button. This will take you over to your Form1.cs code editor. A new method has been added as so:

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

}
```

31. Add the following line to this new method:

```
this.webBrowser1.Navigate( this.comboBox1.SelectedItem.ToString() );
```

32. Build and Run your application. Selecting an item in the combo box and pressing Go now works! Magic!

33. Change the combo box **DropDownStyle** to DropDownList (what is the difference?).