(Commence in 1st prac, due by 4th prac), Worth 1%

Introduction to Visual Studio and C#/VB.NET

The purpose of this exercise is to familiarise you with the new Visual Studio 2012 integrated development environment and the C# programming language.

Part A: Hello World Console Application

- If you're working at home, you'll first need to install Visual Studio
  - See http://software.scitech.qut.edu.au/msdnaa/guide.html
- Start up Visual Studio.NET:
- Create a new project
  - Either from the Start Screen or by selecting "New … Project" from the File menu.
  - Select "Console Application" from the "Visual C#" tab.
  - Set the Project location to "C:\Temp".
- Place the following code in the main method:

```
Console.WriteLine("Hello World!");
```

- Compile and test you program:
  - Select "Build MyProjectName" from the "Build" menu.
  - Select "Start without Debugging" from the "Debug" menu.

Part B: A Simple Windows Application

- Create a new project (in same solution if you wish)
  - Select "WPFApplication" from the "Visual C#" tab.
- Add a push button to the form:
  - Drag a Button control from the Toolbox to the surface of the form.
    - Select "Toolbox" from the "View" menu if the toolbox isn't already visible.
  - Select the button and change its "Content" to "Increment" in the properties window.
    - Select "Properties Window" from the View menu if it isn't already visible.
  - Change the Name of the button to "IncrementButton" in the properties window.
- Add a textbox control to the form:
  - Drag a textbox control from the Toolbox to the surface of the form.
  - Change its Name to "Count", its "Text" to "0" and "IsReadOnly" to True.
- Add an event handler to the push button
  - Select the push button control and select the "Events" tab in the property window.
  - Double click alongside the "Click" event.
  - Add the following code to the new IncrementButton_Click event handler:

```
Count.Text = (Int32.Parse(Count.Text) + 1).ToString();
```

- Compile and test your program.
- Save the project for later use with Part C.

Part C: Component Development and Use in .NET

- Create a new project (in a new solution).
  - Select "Class Library" from the "Other Languages/**Visual Basic**" tab.
- Rename the class to Counter.
- Add a private data attribute called n:

```
Private n As Integer = 0
```

- Add a public method called Increment:

```
Public Sub Increment()
    n = n + 1
End Sub
```

- Add a property called Count:

```
Public ReadOnly Property Count()
    Get
        Return n
    End Get
End Property
```

- Compile your component.
- Save and close the solution.
- Open the solution from Part B.
- Add a reference to the component just created in Part C.
    - Select the Windows Application project in the Solution Explorer window.
    - Right click the project entry and select “Add Reference …”
    - Select the.NET Framework tab and then select “Browse”.
    - Locate the ClassLibrary dll in the bin directory of the Visual Basic project.
    - Verify that your Visual Basic Class Library has been added to the list of references.
- Edit the code associated with the form.
    - Right click the Form1.cs file in the “Solution Explorer” and select “View Code”.
- Add a data attribute called counter to the Form1 class:

```
private ClassLibrary1.Counter counter = new ClassLibrary1.Counter();
```

- Change the `IncrementButton_Click` event handler to:

```
counter.Increment();
Count.Text = counter.Count.ToString();
```

- Compile and test your program.

Part D: Integrated Development Environment

- Learn how to use the Visual Studio.NET Integrated Development Environment effectively.
    - Explorer the menu options available across the top of the screen.
    - Learn how to use the “Solution Explorer” and the “Class View”
        - Check out the options available by right-clicking various items.
        - Learn how to modify the properties of a project.
    - Learn how to use the interactive debugger.
        - Insert break points by clicking to the left of a source line.
        - Step into and over function calls.
        - View the state of variables in the “Autos”, “Locals” and “Watch” windows.

Part E: Online Documentation

- Learn how to use the online documentation.
    - Goto http://msdn.microsoft.com/library/
    - Start by examining:
        + .NET Development
            + .NET Framework 4.5
                + .NET Framework Class Library
                    + System.Collections Namespaces
                        + System.Collections.Generic
                            + Dictionary Class

What type parameters does the constructor take?

Part F: C# or VB.NET Language Features

- If you are not already a C# expert, write simple test programs to explore some of the following C# language features
    - C# Namespaces, classes and interfaces
    - Value vs. reference types
    - C# Arrays,
    - foreach statements
    - Properties
    - Delegates
    - virtual, new and override
    - I/O
    - Strings
    - in, out and ref parameters.
- If you are already familiar with all of these features in C# then do the exercise for VB.NET instead.