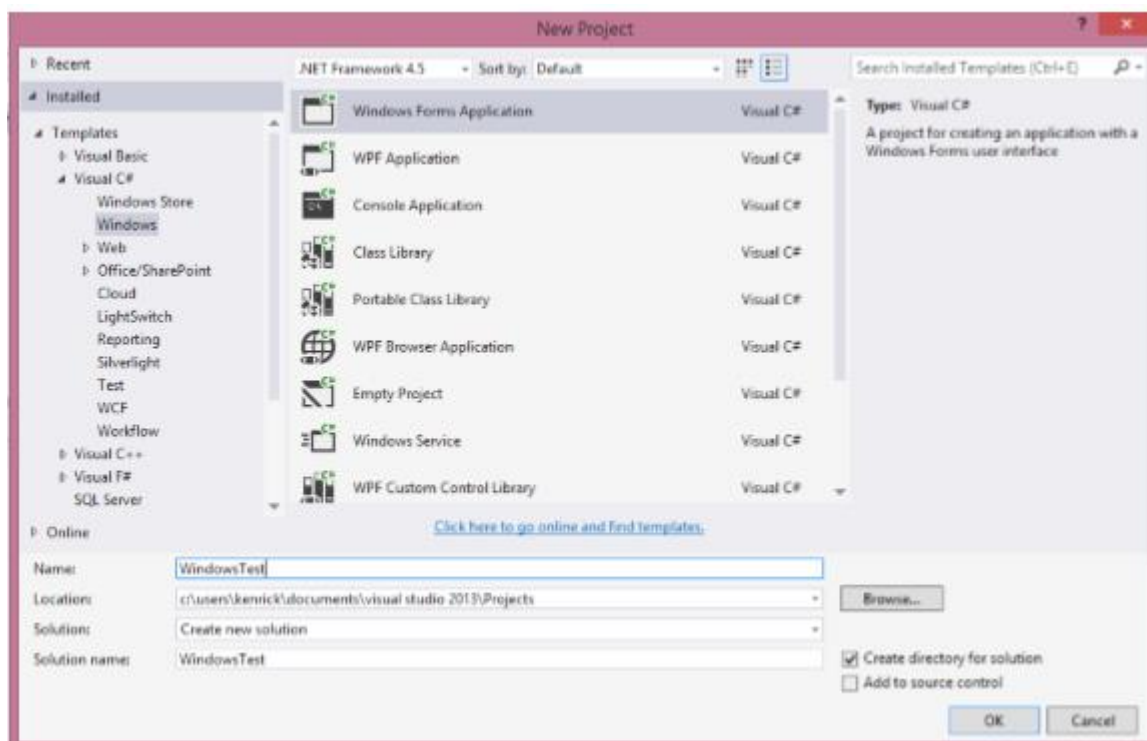


## Creating Windows Forms Applications with Visual Studio and C#

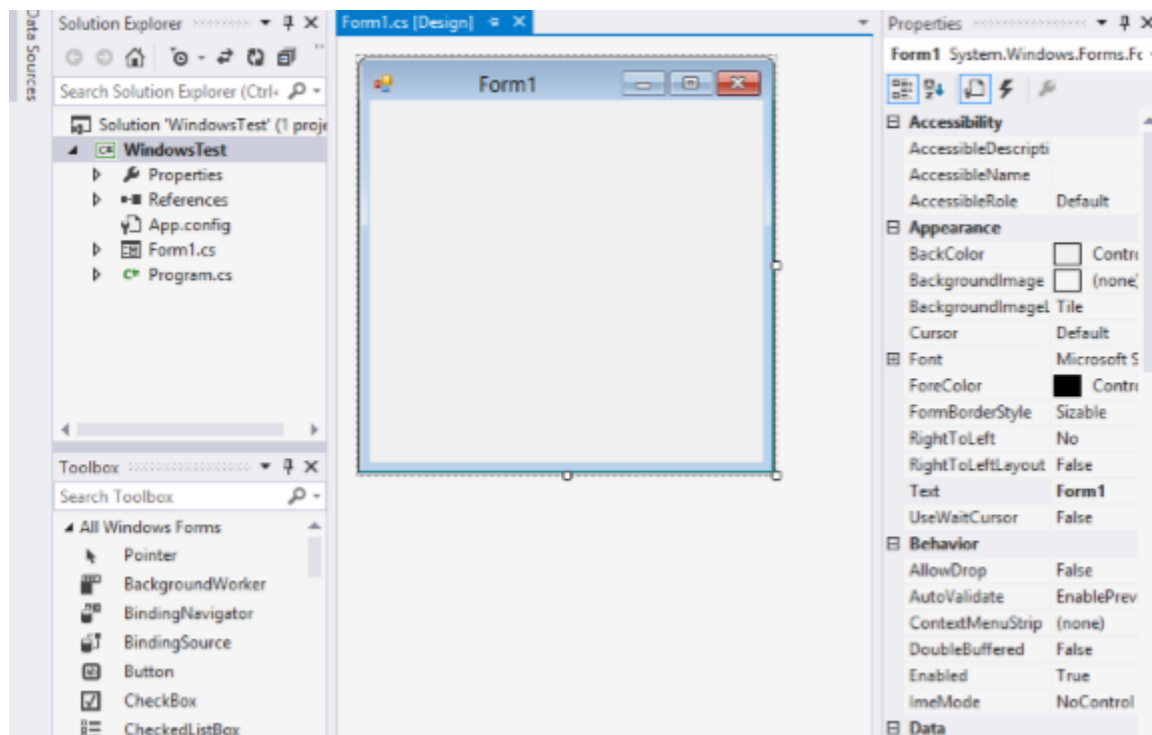
Visual Studio on a Windows platform gives you a multitude of classes to easily create typical Windows GUI applications. If you elect to use these features, currently a C# application will only run on a Windows machine. There is similar functionality for C# GUI applications under Mono. In this lecture we will present just the basics to create some simple Windows apps.

Although it is possible to create windows applications purely in a textual environment, we will instead make use of the Visual Studio interface which lets us graphically drag and drop to determine the layout of our program. The VS environment will then generate the necessary code for us to perform the layout.

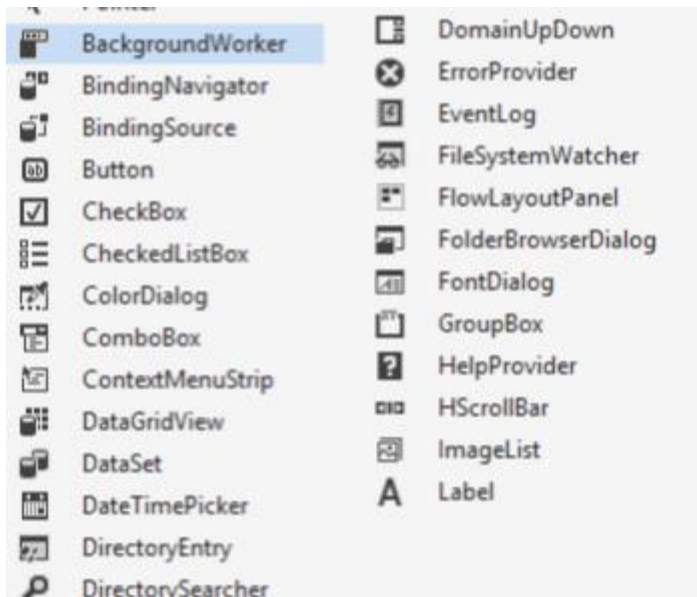


The following screenshots were created in VS 2013. If you have a different version, some windows and/or content may have changed but the basic process is the same. To create a Windows Forms Application, start Visual Studio and create a new Visual C# Project. Make sure you select a Windows Application as the template.

You will be presented with a blank form. This document assumes that you have the Toolbox pane visible on the left of the screen and the Properties on the right. These panes can be moved around and hidden. This pane contains common controls which make up the elements on the form. The layout is shown below:



A control is a component with a visual representation. The Control class in the System.Windows.Forms namespace is the base class for the graphical objects on the screen. All controls are derived from the base control. Examples of controls include:



A Form is a type of ContainerControl. It can contain other controls and we can add controls to it.

In this document we will only discuss a few controls: Buttons, Labels, TextBox, ProgressBar, and PictureBox. A Button is just like it sounds, a button we can press and when it is pressed some code is invoked. A Label is a way to output some text on the form. A TextBox can be used to output data and provide a way for the user to enter text to the program (although a label is better for data that is purely output only). A ProgressBar displays a graphical bar of progress for some slow event. Finally, a PictureBox is used to load and display an image. Please refer to the VS documentation for details about the other controls.

Let's make a simple Windows application that will allow the user to enter a number into a TextBox and press a button. Upon pressing the button we will output if the number is even or odd into a label.

First, select the form by clicking on it. In the Properties Window, you can click and modify various properties of the form. By default, it is called Form1. Scroll through the properties; there are many available to change, such as the size, icon, width, locked, etc. One we can change is the title bar shown with the form. Here

we change the title bar of the form to “Even Odd Tester” by changing the Text property.

