# Exercise 1: .NET Applications

# Goals:

1. Use Google to figure out how to do something new in Visual Studio
2. Create the three basic types of .NET applications

# Assignment:

## Creating “Hello World”

Traditionally, the first application that anyone writes in a new language or environment is called “Hello World”. Its sole functionality is to output the message “Hello, World!” to the user. In this exercise, you will use Visual Studio to create three different “Hello World” applications: a console application, a Windows Forms application, and an ASP.NET web application.

## Testing Your Applications

The instructions below include “test scripts” for each application. A test script is a sequence of steps, with the expected behavior of the application after each step. In future exercises, you will learn how to write your own test scripts and how to automate test scripts. Make sure that your application passes the tests.

## Deploying Your Applications

When you have a working version of your application, and you have tested it from Visual Studio, it is time to deploy your application. Deploying an application is delivering to a (real or simulated) user. For this exercise, deployment of the console and Windows Forms applications will consist of copying the application (Ex1\_Console.exe or Ex1\_WindowsForms.exe) from the “bin/Debug” folder to the top-level “applications” folder.

This takes the application out of the context of Visual Studio, and mimics the environment in which the application will run on a hypothetical end user’s computer. After every time you deploy your application, make sure it still works correctly from the deployment directory!

## Deploying ASP.NET Applications

Deploying an ASP.NET application requires an IIS server, which is beyond the scope of this exercise (even the extra credit). If you’re feeling terribly ambitious, feel free to install IIS on your computer and learn how to deploy to it.

# Discussion:

1. What is accomplished by writing “Hello World” in a new language or environment?
2. What else would you need to know in order to write useful applications?

# Reminders:

1. Commit your code using Git every time you make a change that works.
   1. You can make further changes without worrying about “breaking” your application, since you can always go back to an earlier version.
   2. You will be following the industry best practice of “frequent check-ins”.
   3. You will be “showing your work”, which may allow you to get credit for the assignment even if you are unable to get the tests to pass.
2. Commit your (working) basic implementation before attempting any extra credit.
3. Remember to deploy your applications and test them again in the deploy directory!

# Grading Criteria:

1. Create a console application that says “Hello, World!”
   1. Name the application “Ex1\_Console”.
   2. Create it in the “console app” folder.
   3. Test:
      1. Open a command window
      2. Change the current directory to the “bin/Debug” folder (relative to the base directory, this would be “console app/Ex1\_Console/Ex1\_console/bin/Debug”
      3. Type “Ex1\_Console” and press Enter.
      4. Verify that you see the text “Hello, World!”
   4. After your application passes the tests, deploy the application (the Ex1\_Console.exe file) from the bin/Debug folder into the top-level “applications” folder and re-test
   5. Extra credit 1:
      1. After greeting the user, prompt them to press any key to continue.
      2. Wait for them to press a key before exiting the application.
      3. Test:
         1. In Windows Explorer, double-click on the Ex1\_Console.exe file
         2. Verify that a console window opens.
         3. Verify that it contains the text “Hello, World!”
         4. Verify that it contains the text “Press any key to exit.”
         5. Verify that the command window does not close until you press a key.
         6. Verify that the command window closes when you press a key.
   6. Extra credit 2a:
      1. Expect the user to enter their name on the command line.
      2. Say “Hello, <name>!” instead of “Hello, World!”
      3. Test:
         1. Open a command window.
         2. Change the current directory to the folder containing the application.
         3. Type “Ex1\_Console Bartholomew” and press Enter.
         4. Verify that you see the text “Hello, Bartholomew!”
      4. What happens if you type “Ex1\_Console” and press Enter?
   7. Extra credit 2b:
      1. Allow the user to omit entering their name on the command line. If they omit their name, just print “Hello, World!”
      2. Test – new functionality:
         1. Open a command window to the application folder.
         2. Type “Ex1\_Console” and press Enter.
         3. Verify that you see the text “Hello, World!”
      3. Test – regression test (making sure that existing functionality continues working when you have added new functionality):
         1. In the same directory, type “Ex1\_Console Winifred” and press Enter.
         2. Verify that you see the text “Hello, Winifred!”
2. Create a Windows Forms application that says “Hello, World!”
   1. Name the application “Ex1\_WindowsForms”.
   2. Create it in the “Windows Forms app” folder.
   3. Test:
      1. Double-click on the “Ex1\_WindowsForms.exe” file.
      2. Verify that a window opens containing the text “Hello, World!”
   4. Don’t forget to deploy and re-test!
   5. Extra credit 1:
      1. Prompt the user to enter his or her name.
      2. When the user presses “Enter”, display “Hello, <name>!” instead of “Hello, World!”
      3. Test:
         1. Double-click on the “Ex1\_WindowsForms.exe” file.
         2. Verify that a window opens containing the text “Please enter your name” and a TextBox.
         3. Enter “Bartholomew” and press Enter.
         4. Verify that the window now contains “Hello, Bartholomew!”
   6. Extra credit 2:
      1. Display “Press “Esc” to exit”
      2. When the user presses the “Esc” key, exit the application.
      3. Test:
         1. Double-click on the “Ex1\_WindowsForms.exe” file.
         2. Verify that a window opens containing the text “Press “Esc” to exit”
         3. Press the “Esc” key.
         4. Verify that the window closes.
      4. Regression tests:
         1. Re-run the basic test and the Extra credit 1 test.
3. Create an ASP.NET web application that says “Hello, World!”
   1. Name the application “Ex1\_AspDotNet”
   2. Create it in the “ASP.NET web app” folder.
   3. Test:
      1. Press F5 from Visual Studio to launch your web application.
      2. Verify that an Internet Explorer browser window opens, containing the text “Hello, World!”
      3. If you have changed your default browser to something other than Internet Explorer, verify that one of those browser windows opens, instead.
   4. Extra credit 1:
      1. Prompt the user to enter his or her name.
      2. When the user presses “Enter”, display “Hello, <name>!” instead of “Hello, World!”
      3. Test:
         1. Use the same test script as for the Windows Forms application, Extra credit 1, modified appropriately.