Compiling and Running C# Programs in Windows

(**For running C# on a Mac**, see separate document – "CompileRunGuide_C# - MAC")

Since **C#** was developed by *Microsoft*, it is best if you develop **C#** programs on a *Windows-based* machine, not a *Mac*.

Windows XP

For **Windows XP**, you need to download and install the *.NET Framework* (which includes the **C#** compiler.) Go to: http://www.microsoft.com/en-us/download/details.aspx?id=21. This is the *.NET* Framework Version 3.5. Download and install the framework.

Windows 7 / Windows 8 / Windows 10

Windows 7-10 already has the .NET framework installed. You don't need to download anything!

Setting the Path

First, you need to determine which version of the *.NET framework* is installed. Go to:

```
C:\WINDOWS\Microsoft.NET\Framework
```

You should see a folder in there that says "v3.5" or "v4.0.30319" (or both). Go into the folder **with the highest number** after the "v". To copy YOUR 'Framework path', right-click in the directory window at the top and choose **'Copy address as Text'**

Like with Java, you will add this directory to your ***path*** environment variable by doing the following:

- Open the `Control Panel`
- Open `System and Security`
- Open `System`
- Click `Advanced (System Settings)`
- Click `Environment Variables`
- Find the variable `Path` and click on it (under the *System variables* window in Win 10)
- Press `Edit`

`In Windows XP/7/8` (see next page for Windows 10)

- Hit the right arrow key to move to the END of the `Path` value (*make sure your current path is **not** highlighted or it will REPLACE your current path*!). Type a semicolon, right click and PASTE the path directory you *copied as text* then type another semicolon and type a period. It should resemble something like…

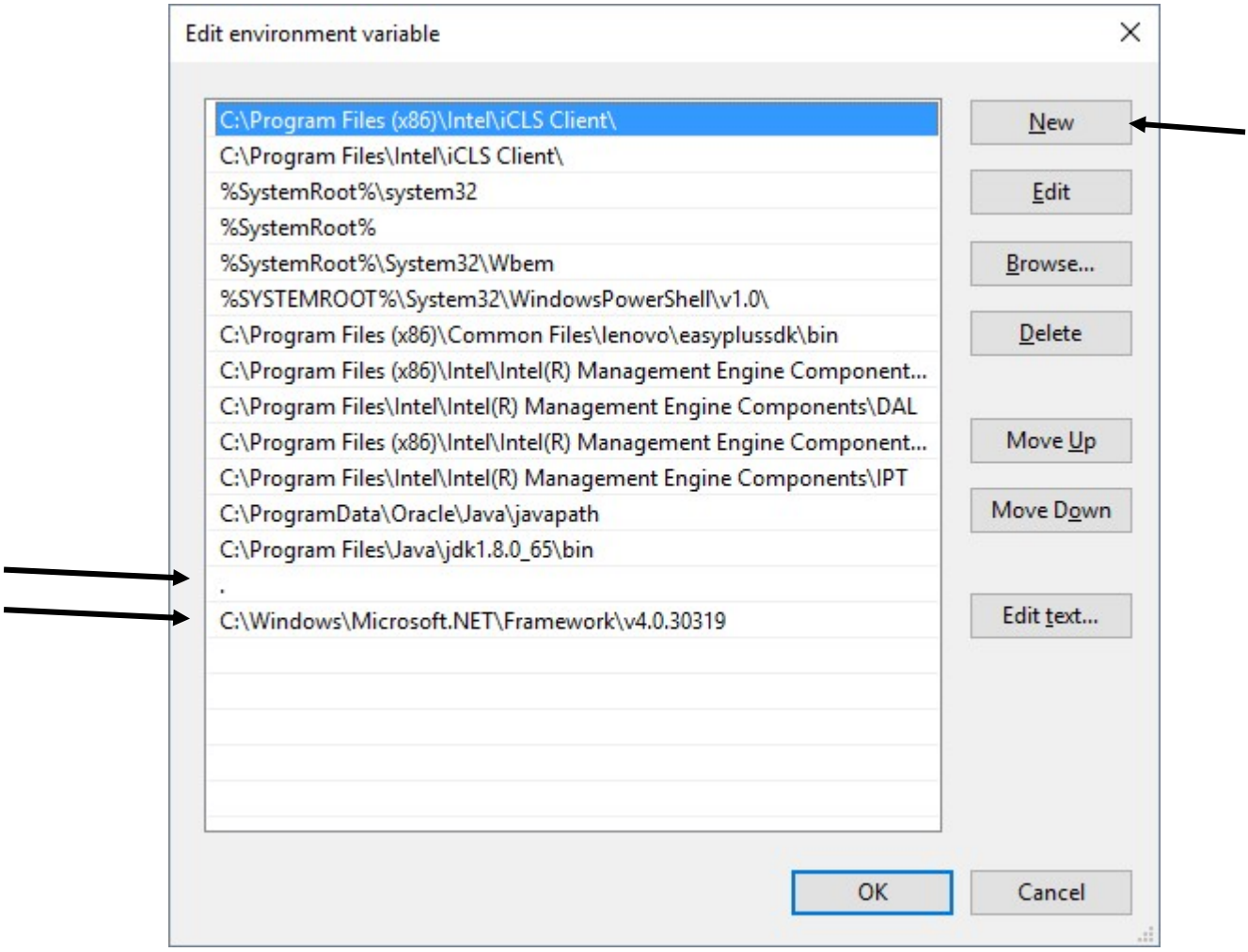
  `;C:\WINDOWS\Microsoft.NET\Framework\v3.5;.`

  (Replace *v3.5* with whatever folder you found)

- Press `OK` twice

In Windows 10

- Click on NEW to add to the path. PASTE the path directory you ***copied as text*** then click on OK. If there is not a period also listed, repeat the steps to add a '.' to the Path (which represents the current directory). It should resemble something like… (order doesn't matter…)



- Press `OK` twice

To verify this was added correctly, open a command window and type in **SET**. You should see a PATH variable with the above path correctly added. If not, your program will NOT compile correctly.

Compiling from the Command-Line

Any text editor can be used to create a C# program. Just save with the file extension .cs

To compile a **single-class** C# program stored in the file **`Hello.cs`**, first open a command window:

- Press "`Start`"
- Type **`cmd`** in the search

Change directories (use the **`cd`** command) to the folder where **`Hello.cs`** is stored. **For example**, if it was stored in a folder called 'temp' on my desktop, I would type: **cd \Users\DLang\Desktop\temp**

Type ***DIR*** at the system prompt to see a list of files stored in the directory (folder).

To compile your program, type: **`csc Hello.cs`**

If everything works correctly, it will compile your program into the executable file **`Hello.exe`**, which will be stored in the same directory as **`Hello.cs`**. If you made any mistakes in your program, the compiler will display error messages describing the problems.

If you want to compile a C# program that has several classes, type: **`csc *.cs`**
`(where * is an asterisk)`

This will compile ALL your classes and will generate an **`.exe`** file with the same name as the file containing the *Main* method.

Running a program from the Command-Line

Suppose you just compiled your **`Hello.cs`** file and have created the executable **`Hello.exe`**. To run your program, simply type: **`Hello`** in the command window (again, after changing directories to the folder where `Hello.exe` is located). If you compiled a program with several files, then the executable has the same name as the class with the *`Main`* method.

Note: You can also **double-click** on the **Hello.exe** file from your desktop to execute it.

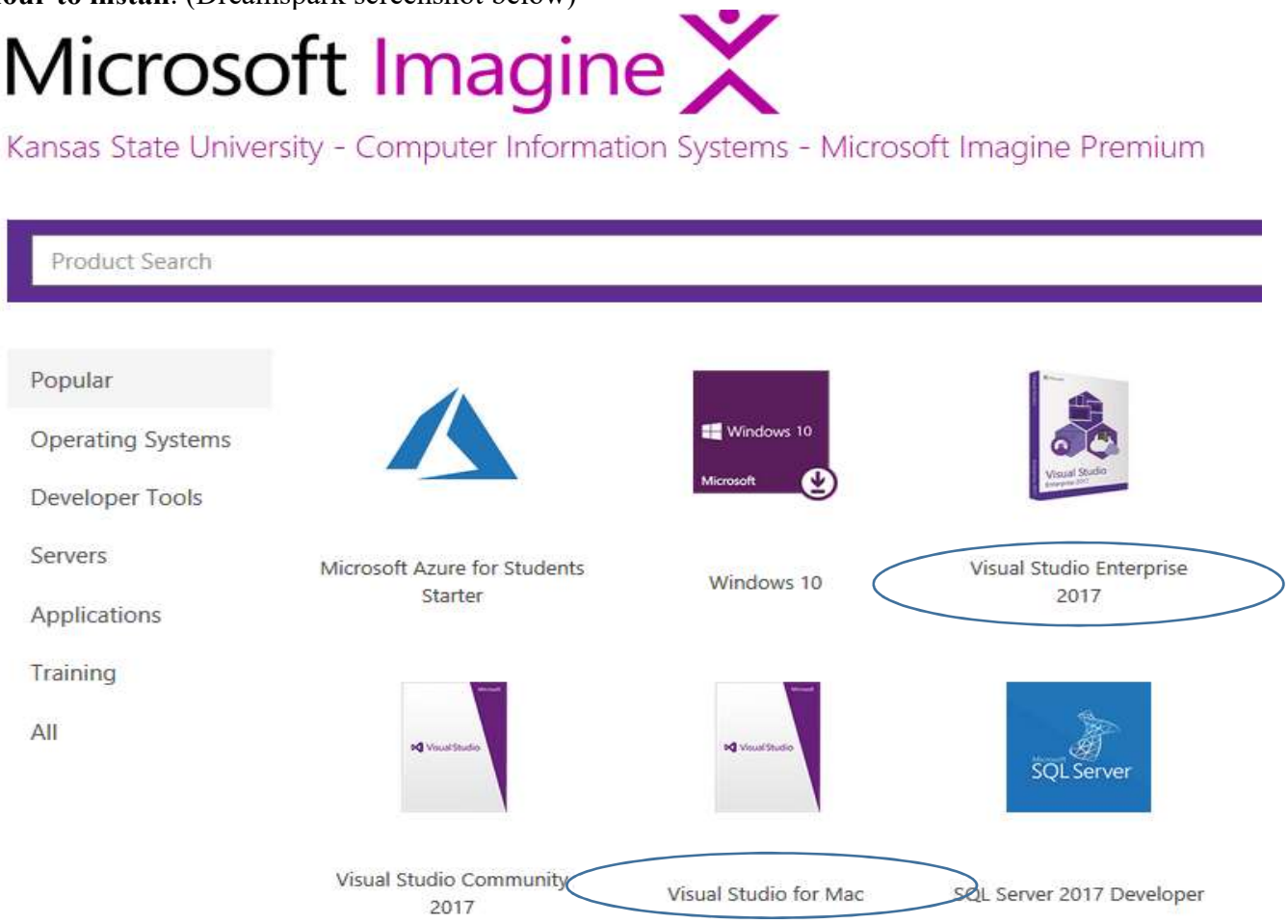
Compiling/Running C# GUI-based applications using Visual Studio

Visual Studio IDE

The link below takes you to **Microsoft's Visual Studio homepage** where you can download, for FREE, the **Community Version** of Visual Studio (VS). Like Eclipse, this is a full-featured IDE that allows you to edit, compile and run your C# programs. Click on **Download** then run the install file once it downloads. https://www.visualstudio.com/downloads/download-visual-studio-vs

Although the **Community** version of VS is more than adequate for this course, **through your CS account you can get the Enterprise version of VS 2017 FREE through the MSDN Academic Alliance program (DreamSpark)**. At the very beginning of this semester, you should have gotten an e-mail from the system administrator about this program with the subject line **Dreamspark accounts for Microsoft and VMWare**, which provided information on accessing the software. Please look for this email first (search for *DreamSpark* in your inbox). If you can't find it and want the Enterprise version of Visual Studio, you can e-mail the system administrators at help@cs.ksu.edu.

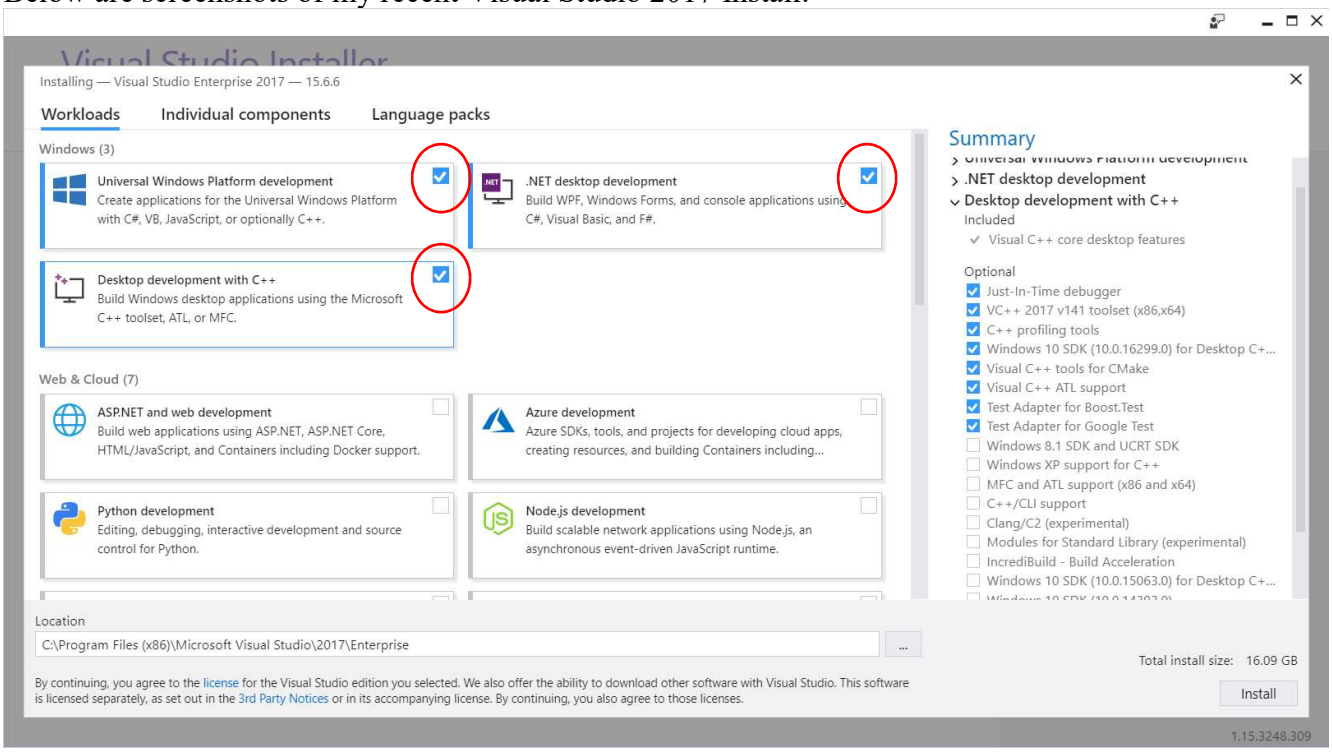
You can also find info on download free software on the CS Support Page (https://support.cs.ksu.edu) …look under the Top FAQ "*I need some software by Microsoft*…". Be aware that it can **take up to an hour to install**. (Dreamspark screenshot below)



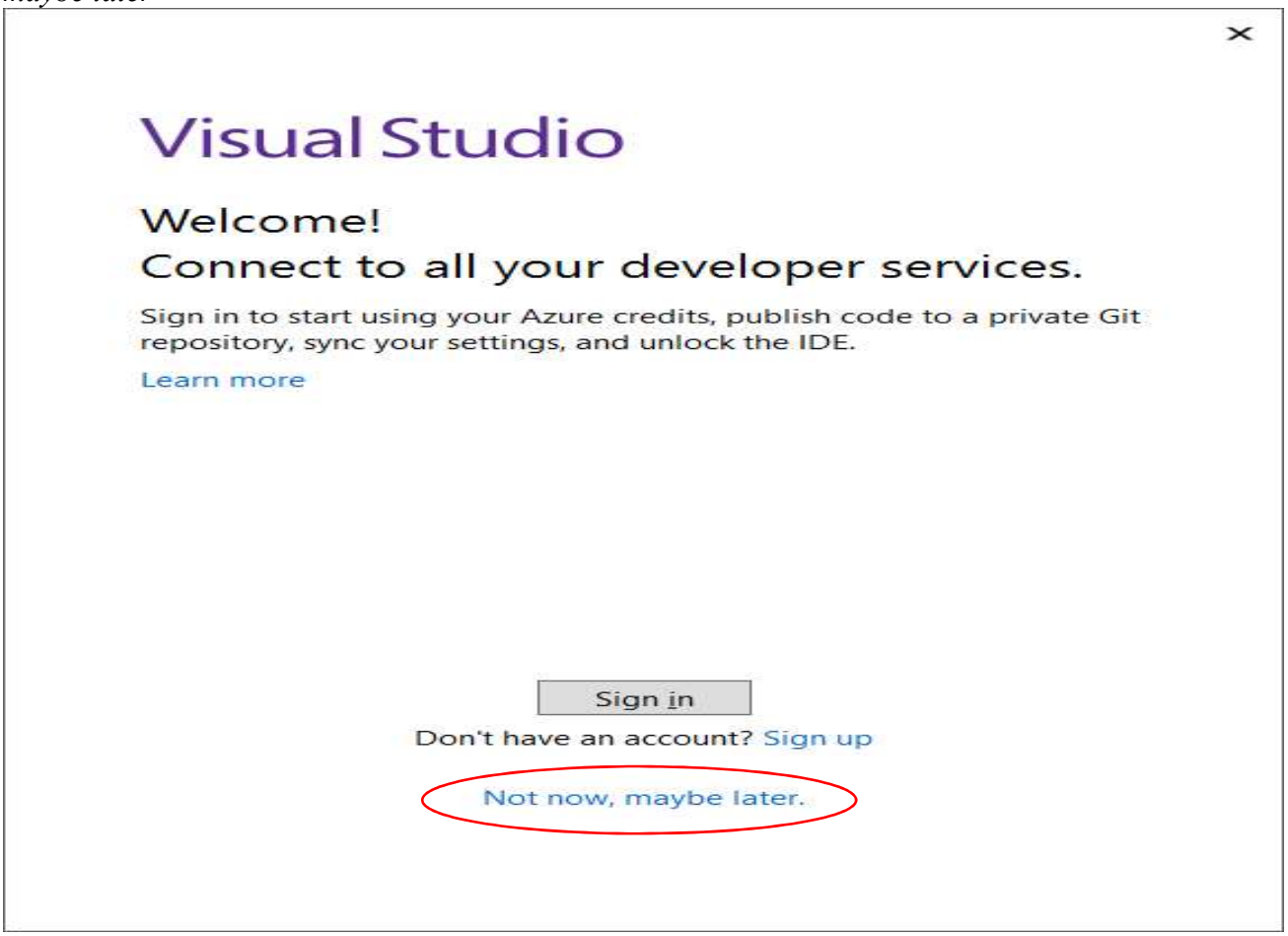
(If the Enterprise edition is downloaded as an **.iso** file, go to the link below for more info: www.techsoup.org/support/articles-and-how-tos/what-do-after-downloading-iso-files-from-microsoft)

FYI: The link below shows a comparison of features between different VS releases.
https://www.visualstudio.com/vs/compare/

Below are screenshots of my recent Visual Studio 2017 Install:

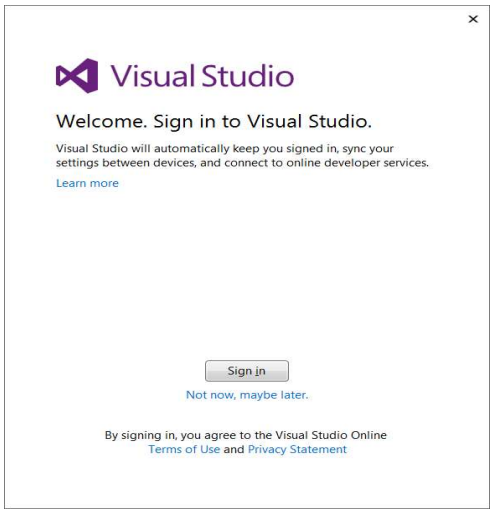


When you first open Visual Studio, this will appear. Feel free to skip ahead by clicking on *"Not now, maybe later"*



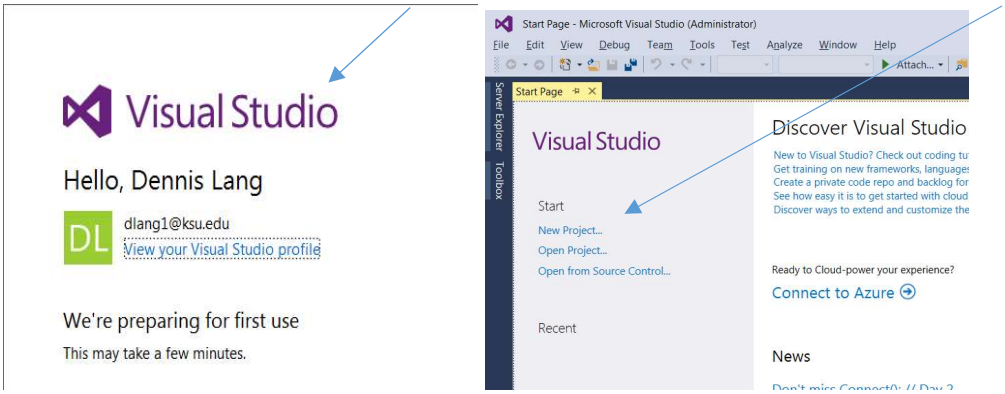
Creating a Project in Visual Studio (VS)

Creating a project (solution) in Visual Studio is very similar to creating a project in Eclipse. You will do this in your final lab session but the directions below can serve as a useful guide. The directions below are for ***Visual Studio (VS) 2015 Community Version***. Creating a project in other versions of VS are very similar. After installing, when you first run VS 2015, the following appears:



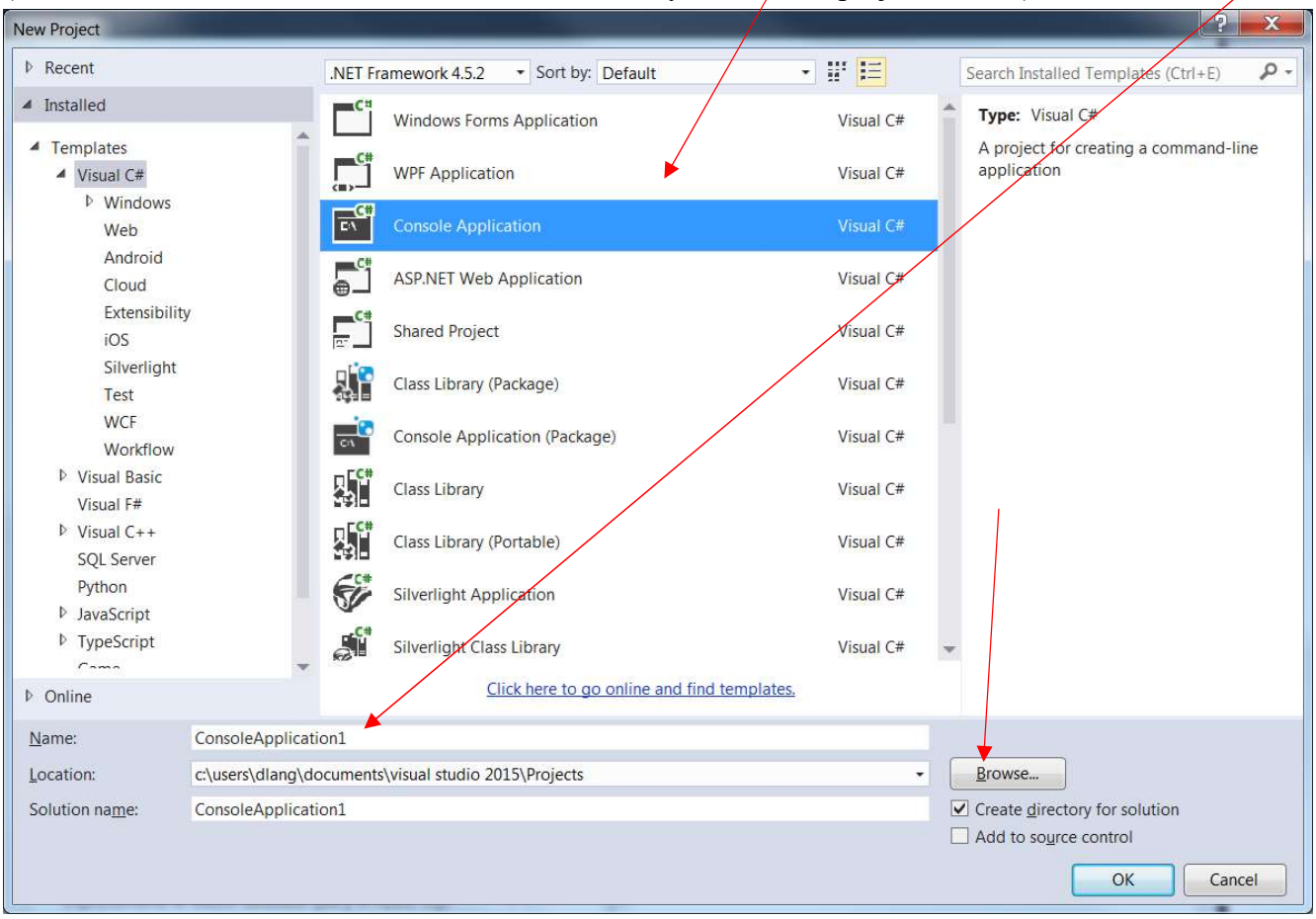
Either click "*Not now. Maybe Later*" or click "*Sign in*" – ***if signing in***, use your **ksu.edu email address** – it will then prompt for your eID and Password.

It will then show the following screen and the program will open. Click on "*New Project*"



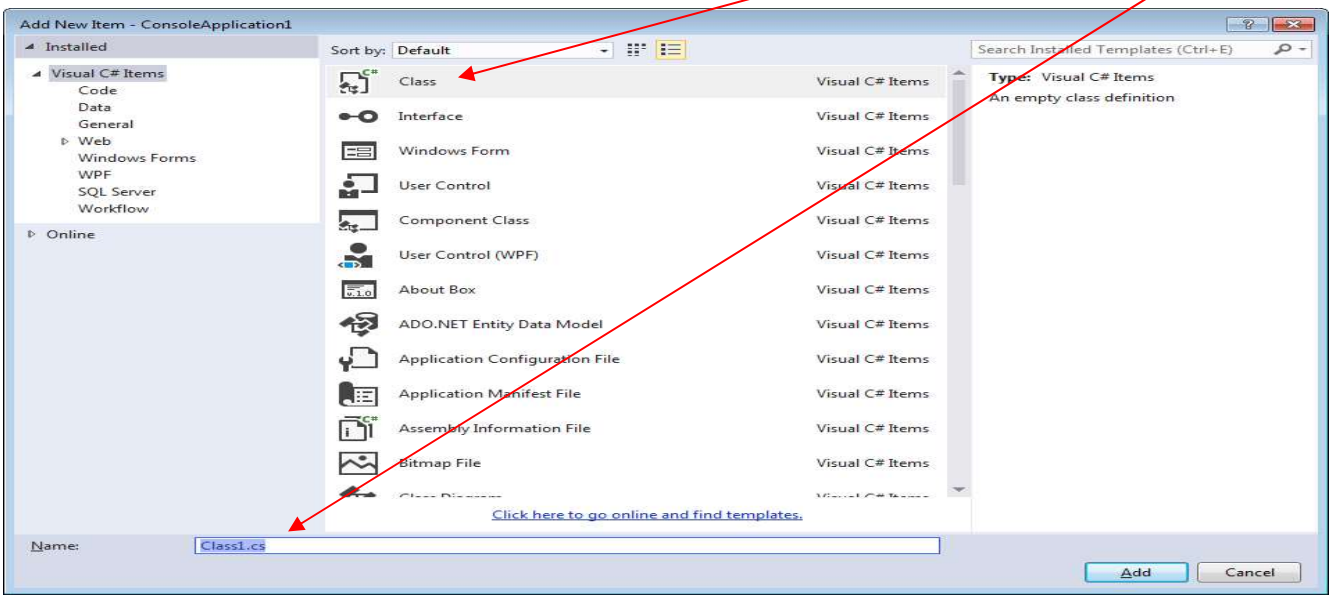
Again, working with ***Visual Studio*** will be discussed in **Lecture** *and* in **Lab** during **Week 13**, but here is a brief summary.

To create a Console-based application, click on "*Console Application*" and give the project a name (make sure and browse to the location first in which you want the project to save)



Select "**OK**" to create the project. Class "***Program.cs***" is created and you are ready to start coding. **To compile**, choose "**Build – Build Solution**". **To run**, click on the green **Start** button. After the program finishes running, the console window *auto closes* and you may not be able to see your output. If this happens, run you program using "**Debug-Start without Debugging**".

To add a class to your project, choose **"Project-Add Class"** then select the top choice, enter the class name and click on **Add**.



When you do this, the code for the new class will appear in the screen in which you can complete the class code. Repeat this process for every class you want in your program.

If you wish to run your code using the green **Start** button and have the console screen hold until your press a key, you can **add the following two lines of code** at the end of your program:

```
Console.Write("Press any key to end");
Console.ReadKey();
```

To create a GUI-based application, click on "*Windows Forms Applications*" and give the project a name (make sure and browse to the location first in which you want the project to save). **This will be demonstrated in lecture and done in your final Lab.**