WASHINGTON STATE UNIVERSITY
World Class. Face to Face.

CptS 122 – Data Structures

Lab 1: Review of Problem Solving with C and Introduction to Microsoft Visual Studio 2015

Assigned: Wednesday, June 7, 2017
Due: At the end of the lab session

I. Learner Objectives:

At the conclusion of this programming assignment, participants should be able to:

- Analyze a basic set of requirements for a problem
- Compose a small C language program
- Compile a C program using Microsoft Visual Studio 2015
- Create test cases for a program
- Apply and implement arrays and strings in C
- Apply and implement recursion in C

II. Prerequisites:

Before starting this programming assignment, participants should be able to:

- Access Microsoft Visual Studio 2015 Integrated Development Environment (IDE)
- Apply basic problem solving strategies
- Design and implement small programs in any language

III. Overview & Requirements:

Welcome to CptS 122 – Data Structure's first lab! Labs are constructed to be hands-on. So be ready to get your hands dirty!

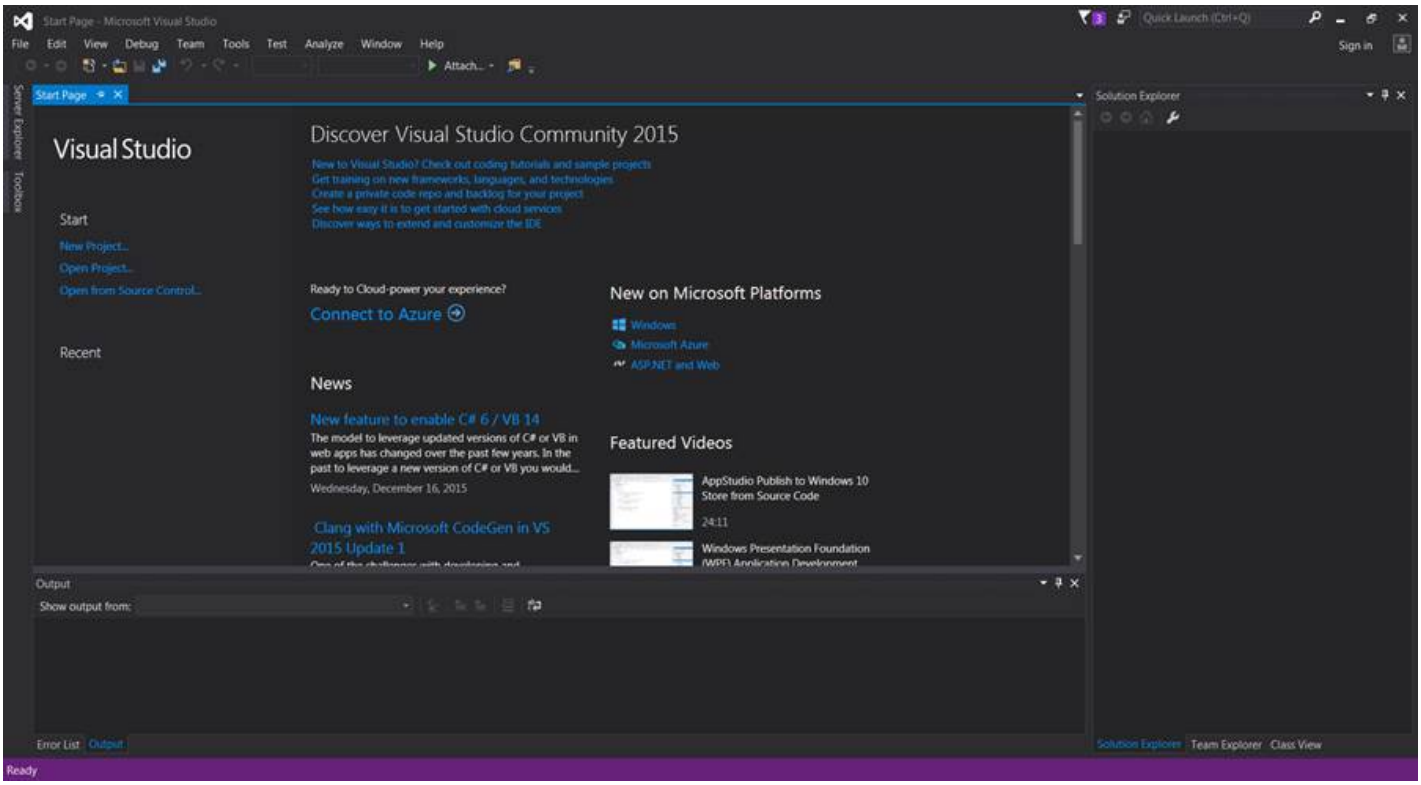
This lab, along with your TA, will help you navigate through Microsoft Visual Studio 2015 Integrated Development Environment (IDE). You will learn how to setup a console application. You will take advantage of the text editor, compiler, linker, and loader that is built into Visual Studio to construct, execute, and test small C programs that solve basic mathematical problems.

Labs are held in a "closed" environment such that you may ask your TA questions. Please use your TAs knowledge to your advantage. You are required to move at the pace set forth by your TA. Please help other students in need when you are finished with a task. You may work in pairs if you wish. However, I encourage you to compose your own solution to each problem. Have a great time! Labs are a vital part to your education in CptS 122 so work diligently.

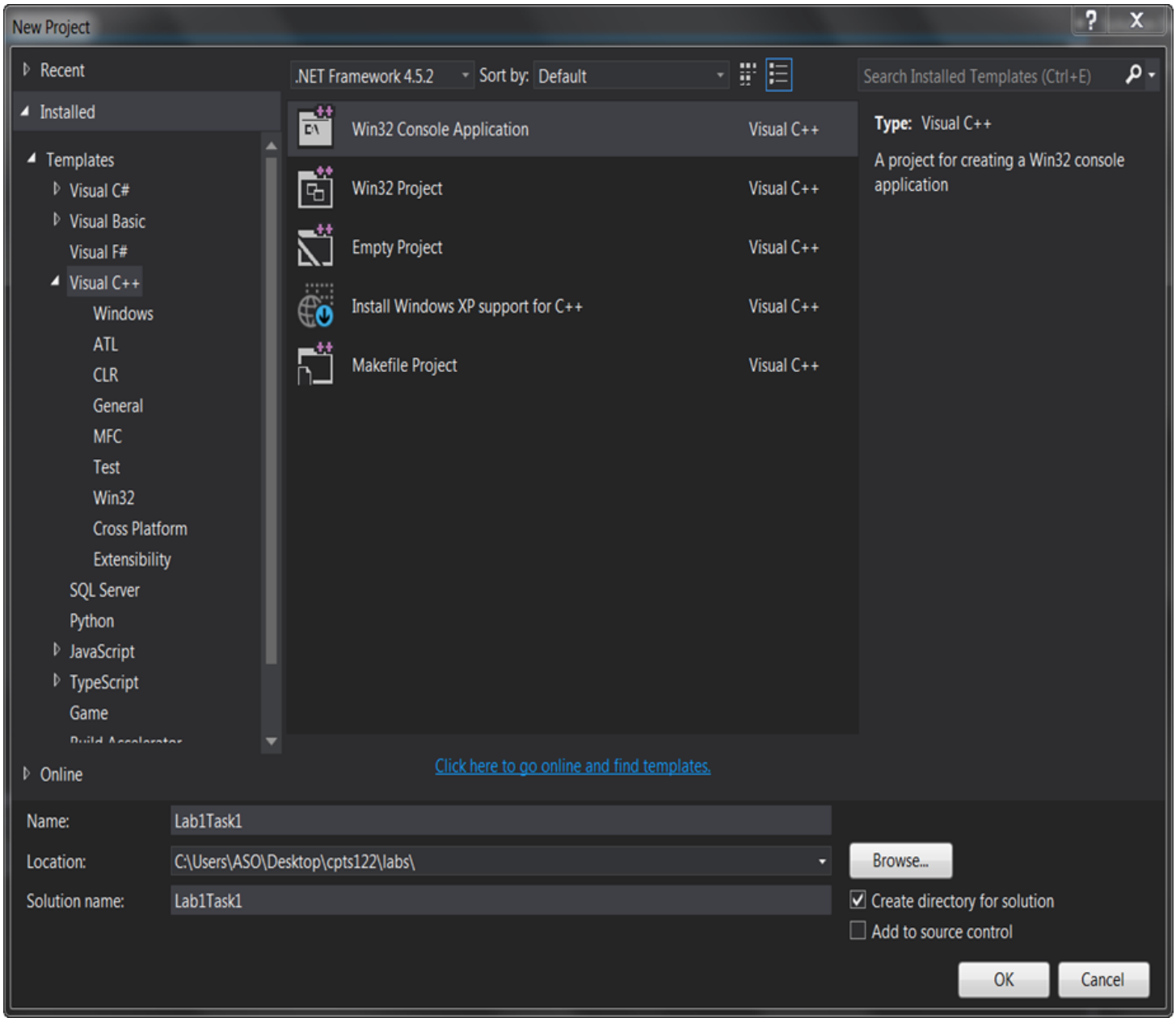
Task 1: Develop a small hello world program utilizing the Visual Studio development environment. All of the applications that we will build in this class will be Console applications. Console applications run in a command window, and have purely textual input and output.

Perform the following steps to build a Console application:

1. Create a folder in an appropriate drive on your computer (use the Z: if you are using with the lab machines), for this class, named cpts122. This is where all programs that you create for this class will be saved.
2. From the start menu select programs, Visual Studio 2015 --> Visual Studio 2015 (or equivalently click on the Visual Studio 2015 icon on the desktop if one exists). This will start the visual studio integrated development environment (IDE). The first time you run the program you will want to select the "General" default environment. The IDE start page is shown below.

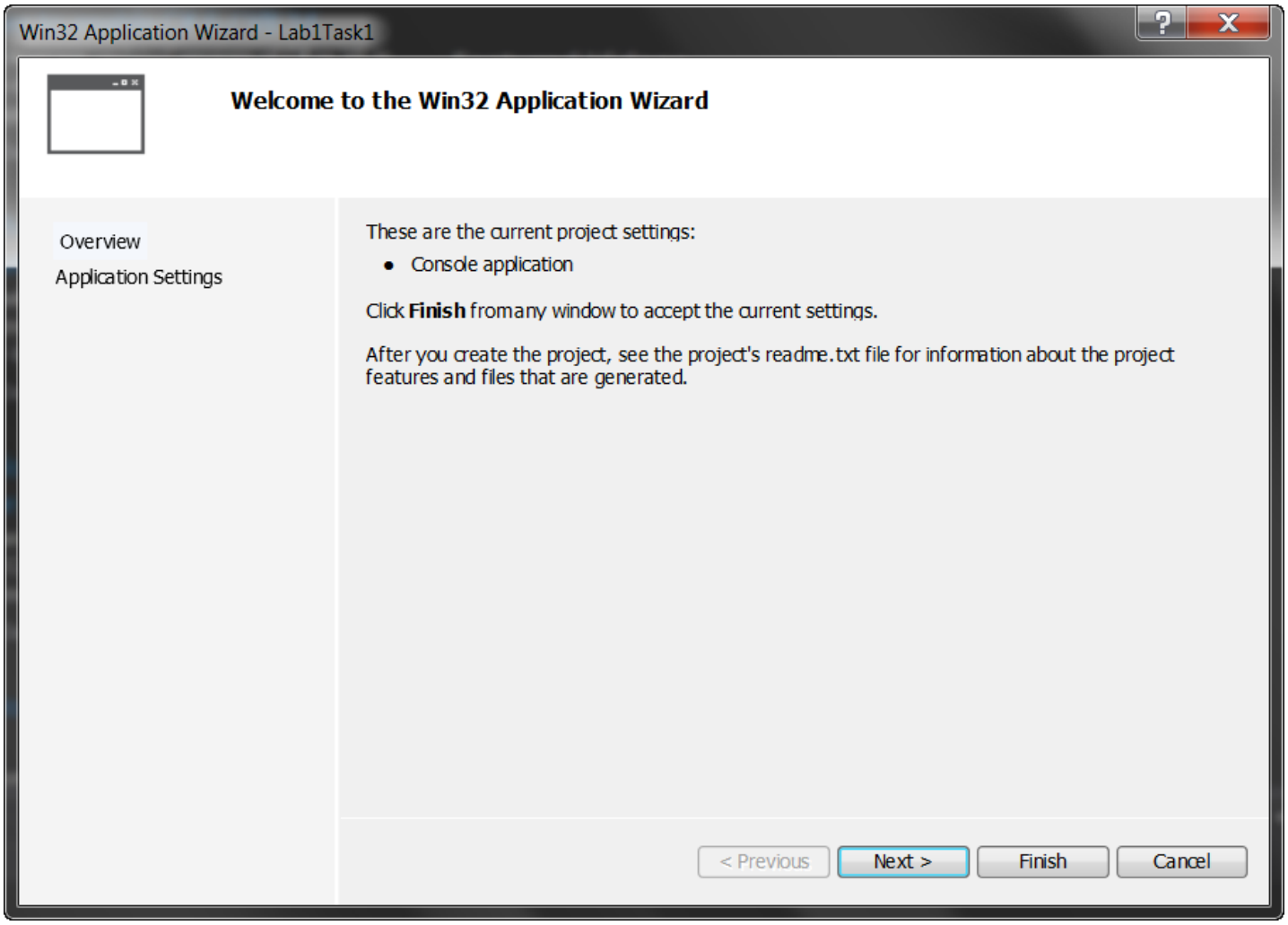


From the “File” menu select “New” and then “Project”. This will bring up the New Project window shown below.

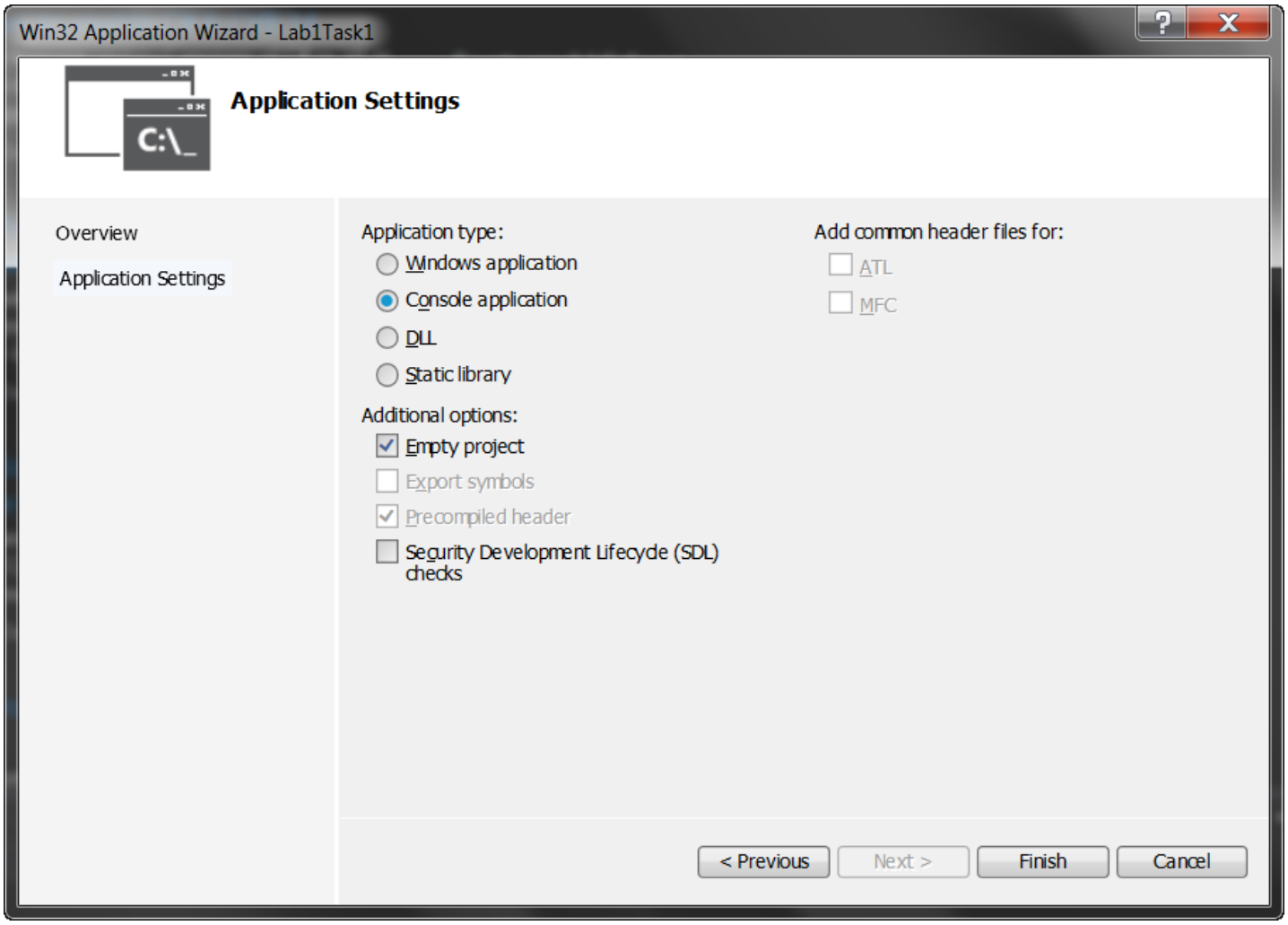


In this window, select “Visual C++” projects on the left and then select “Win32 Console Application” on the right. (See the gray shaded entries above). Next, give the project a name. A naming scheme will make it easier to find different solutions and projects later. At this time, name this project “Lab1Task1”. You also need to enter the Location; it should be “Z:\cpts122” or another location that you like to store project files. Note: if you are using the lab machines, be sure not to store any files on the C: drive. These files will automatically be erased. You can use the “Browse” button or type it in directly. When these are correctly completed, click the “OK” button.

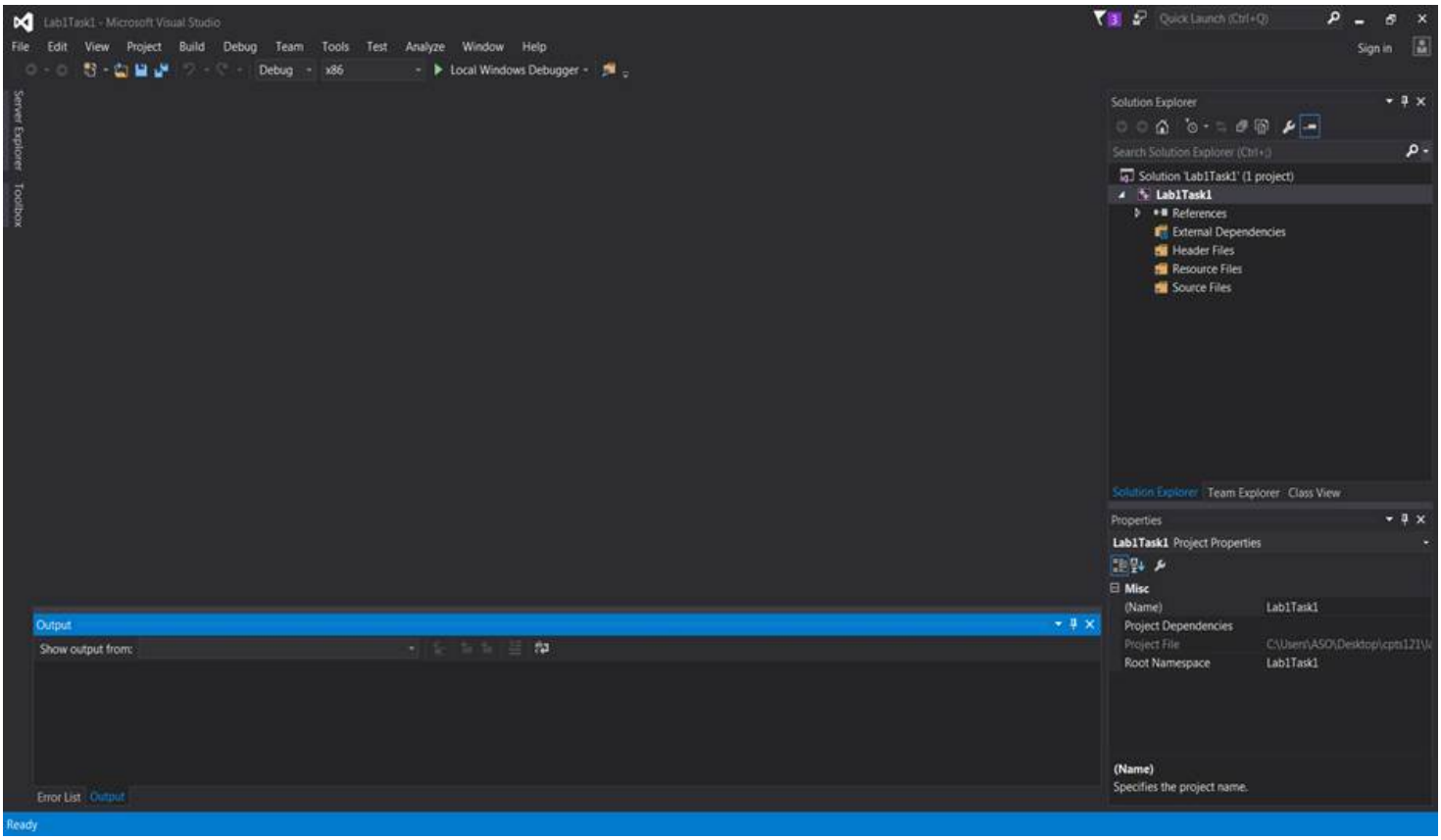
You should now see the Win32 Application Wizard Window shown below, in that window click on the “Next >” button.



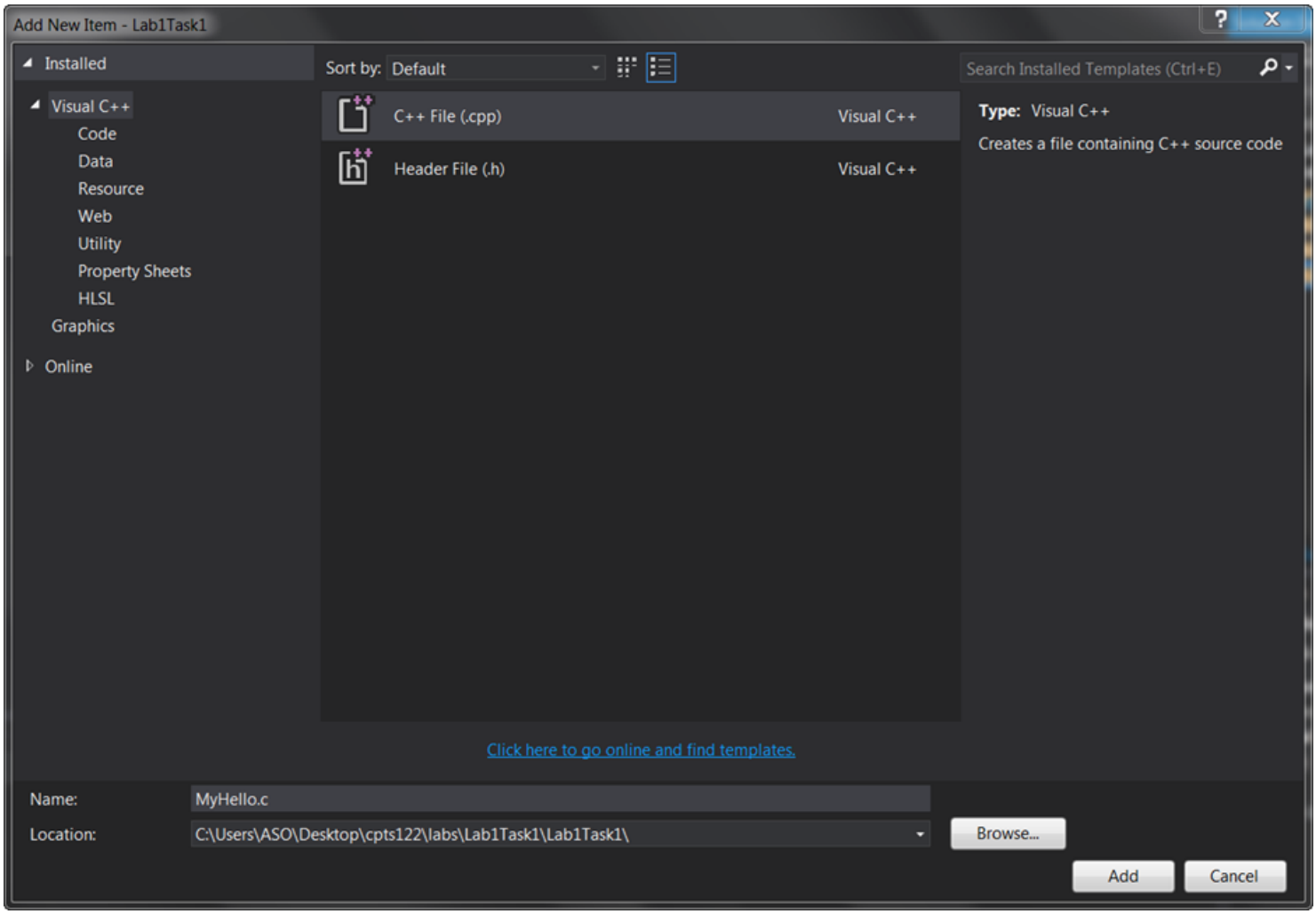
You will see the Wizard window change to provide several choices. Select the “Console application” (should be the default application type already) and “Empty project” as shown in the screen image below. Be sure to deselect the “Security Development Lifecycle checks” option for now. Then click the “Finish” button.



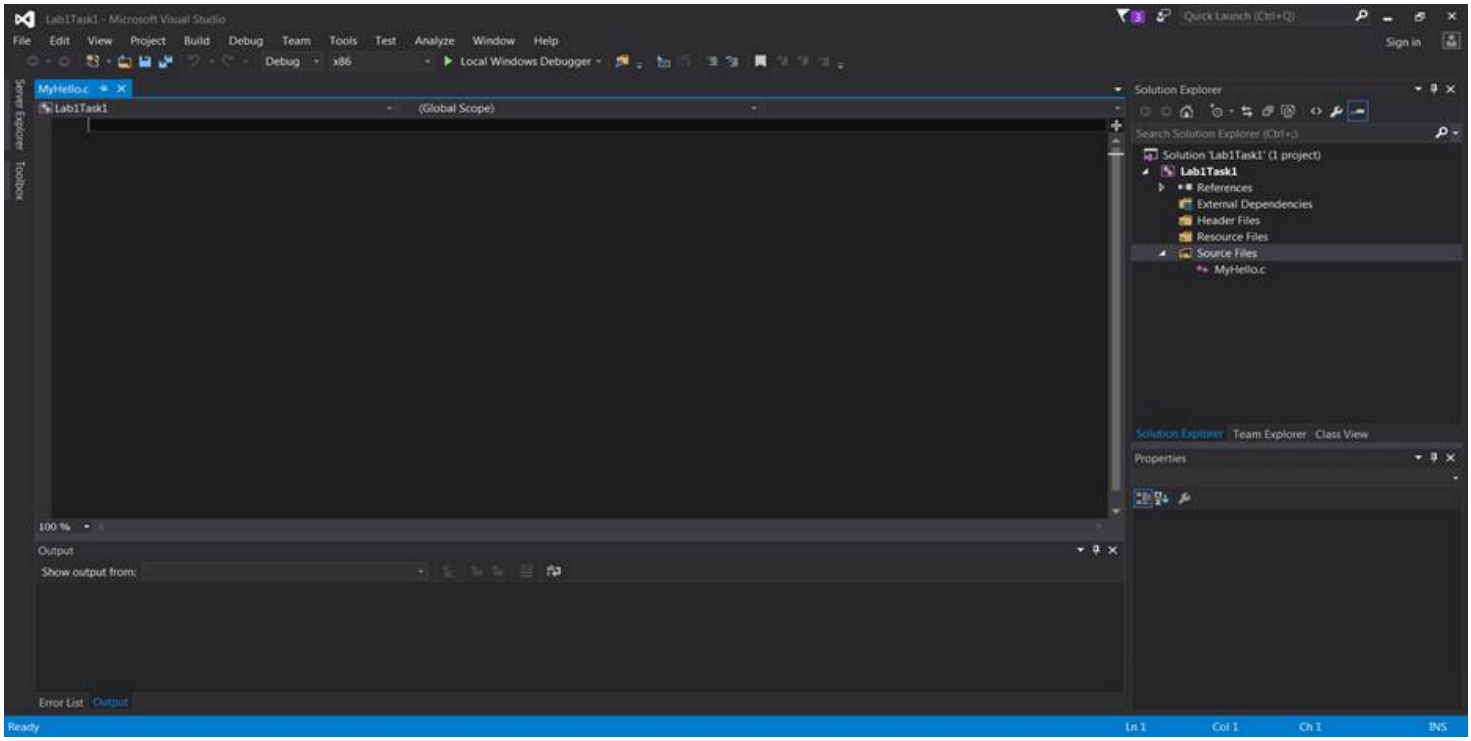
The IDE Window will change a little bit now. There is now information on the right hand side of the window in the Solution Explorer.



You now need to create a new file for your program. Do this by right clicking on the “Source Files” folder in the “Solution Explorer”, in the Visual Studio IDE, and selecting “Add” --> “New Item”. Give your file a name in the filename text field. Name the file MyHello.c. Make sure that you give your file the .c extension indicating that you would like the C compiler to be invoked once you compile your program. If you do not use the .c extension, the source file will default to a .cpp extension. This will cause the C++ compiler to be invoked. Next click “Add”.



When you select “Add”, you will see a file edit window come up inside the IDE with the name MyHello.c. You can enter your Hello World program here.



Make sure that for every program that you write, you have the following comment header block at the beginning of your source code file:

```
/*****************************************************************************************
* Programmer: Your Name                                                                  *
* Class: CptS 122, Summer 2017; Lab Section X                                            *
* Programming Assignment: Lab1Task1                                                      *
* Date: June 7, 2017                                                                     *
* Description: This program prints out a simple "hello world" message.                   *
*****************************************************************************************/
```

For this first program you will only need one executable C statement to output a message. Your program should output "Hello World, I am in CptS 122!" with a printf ( ) statement.

Once you have entered the code for your simple hello world program you will build it from the Solution Explorer. Place the mouse pointer over Lab1Task1 in the solution explorer. Right click and choose “Build”. If you correctly entered the program source code you should see a "Build succeeded" message in the Build Window. If you had an error you will have a failed message in the Build window and a fatal error in the Task List Window. Correct your mistakes and try again.

To run your program when it has successfully built, hit Ctrl-F5 or run it from the “Debug” tab in the IDE.

When you have successfully ran your hello world program, show your TA. Once you have completed this you may close this workspace under the file menu. Close this solution and create a new one for the next task. You should be able to move on to the rest of tasks when your TA instructs you to do so!

Task 2: Design, implement, compile, and test C solutions to the following problems. Once you have completed a problem, demonstrate your solution to your TA.

a. Write a function that recursively reverses a string. Recall, a recursive function is a function that calls itself. These functions have at least one base or simple case and at least one recursive step. The base case(s) have known solutions. As the function is called, the problem is broken down into simpler parts that are closer to the base case.

b. Write a function that integrates or merges two unsorted strings into sorted order. You will need to provide multiple solutions.
   i. Solution 1: Merge items into a third, fixed-sized array.
   ii. Solution 2: Merge items into a third, dynamically allocated (heap) array, which grows as each item is inserted.
   iii. Solution 3: Merge items without the use of a third array.

c. Write a program which reads characters from a file and determines the number of each character, and the line of which each one is found, in the file. Assume the file consists of alphabetic characters only. You may NOT use if statements to "count" each character.

IV. Submitting Labs:

- You are not required to submit your lab solutions. However, you should keep them in a folder that you may continue to access throughout the semester. You should not store your solutions to the local C: drive on the EME 120/128 machines. These files are erased on a daily basis.

V. Grading Guidelines:

- This lab is worth 10 points. Your lab grade is assigned based on completeness and effort. To receive full credit for the lab you must show up on time and continue to work on the problems until the TA has dismissed you.