

## CptS 121 - Program Design and Development

### Lab 1: Introduction to Microsoft Visual Studio 2012 and Problem Solving with C

**Assigned:** Week of January 20, 2014

**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
- Declare variables
- Apply C data types and associated mathematical operators
- Comment a program according to class standards
- Logically order sequential C statements to solve small problems
- Compose a small C language program
- Compile a C program using Microsoft Visual Studio 2012
- Execute a program
- Create basic test cases for a program

#### II. Prerequisites:

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

- Access Microsoft Visual Studio 2012 Integrated Development Environment (IDE)
- Apply basic problem solving strategies
- Summarize key language elements of C including:
  1. Preprocessor directives
  2. Standard and user-defined identifiers
  3. Comments
  4. Punctuation
  5. Operators

#### III. Overview & Requirements:

Welcome to CptS 121 - Program Design and Development'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 2012 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.

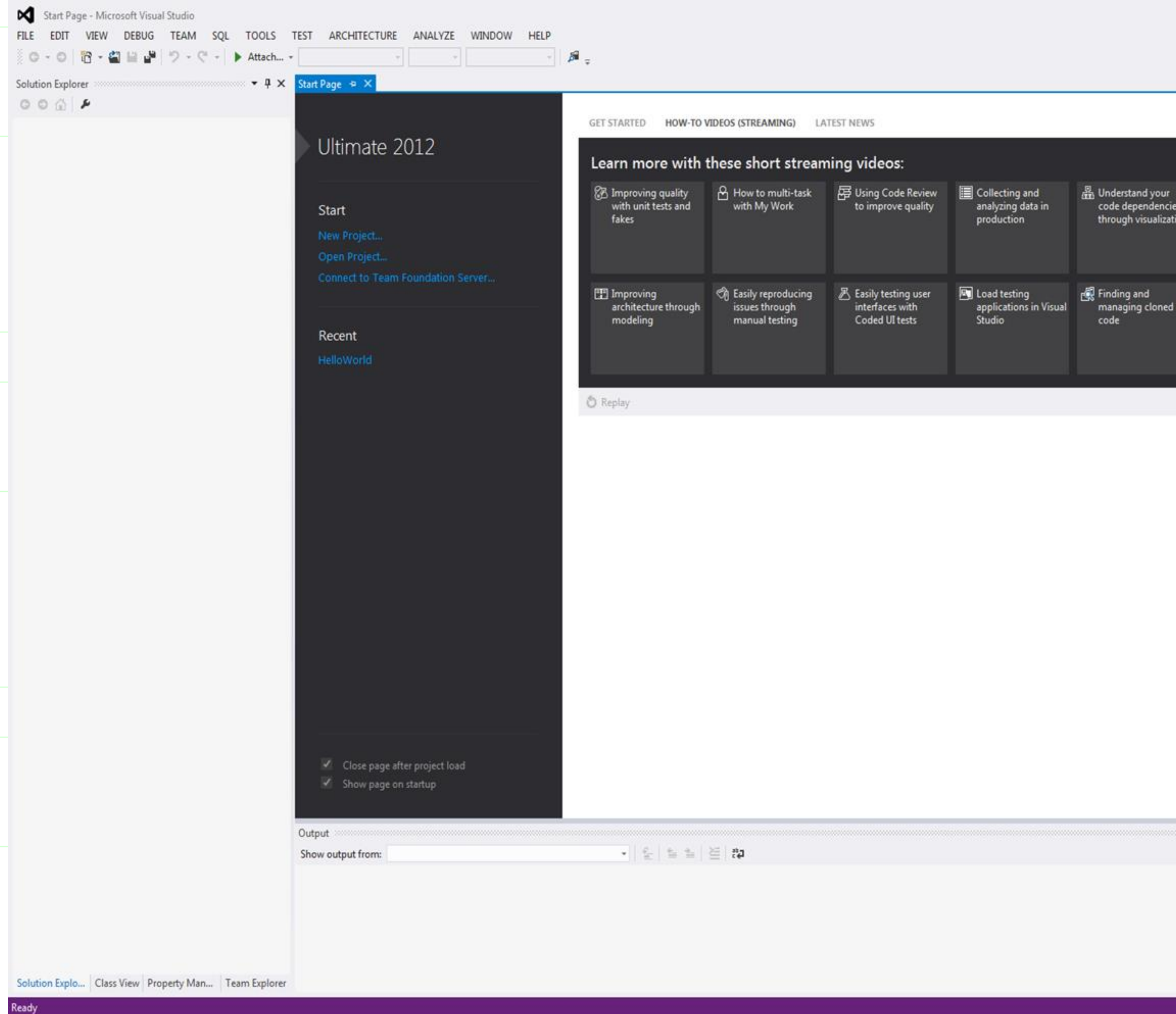
Labs are held in a "closed" environment such that you may ask your TA questions. Please use your TAs knowledge to your advantage. You are required to move at the pace set forth by your TA. Please help other students in need when you are finished with a task. You may work in pairs if you wish. However, 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 121 so work diligently.

**Task 0:** Before you get started with using Visual Studio, please fill-out the survey found at: <http://survey.osble.org/index.php?sid=83956&lang=en>. The TA will

**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.

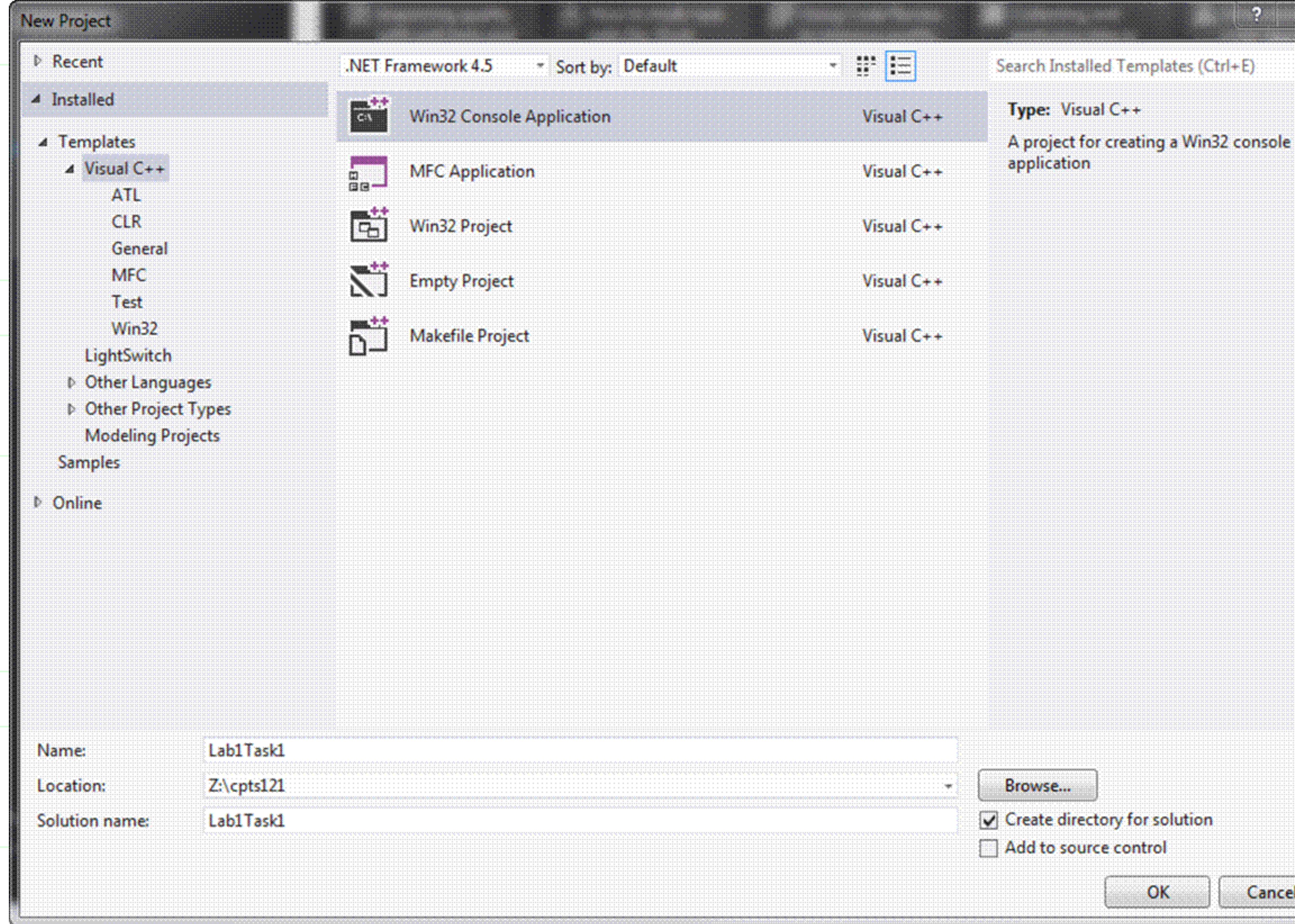
Perform the following steps to build a Console application:

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 cpts121. This is where all programs that you create for this class will be saved.
2. From the start menu select programs, Microsoft Visual Studio 2012 --> Visual Studio 2012 (or equivalently click on the Microsoft Visual Studio 2012 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 C++ as your default environment. The IDE start page is shown below. You may have Visual Studio 2012 Professional or Ultimate installed.



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

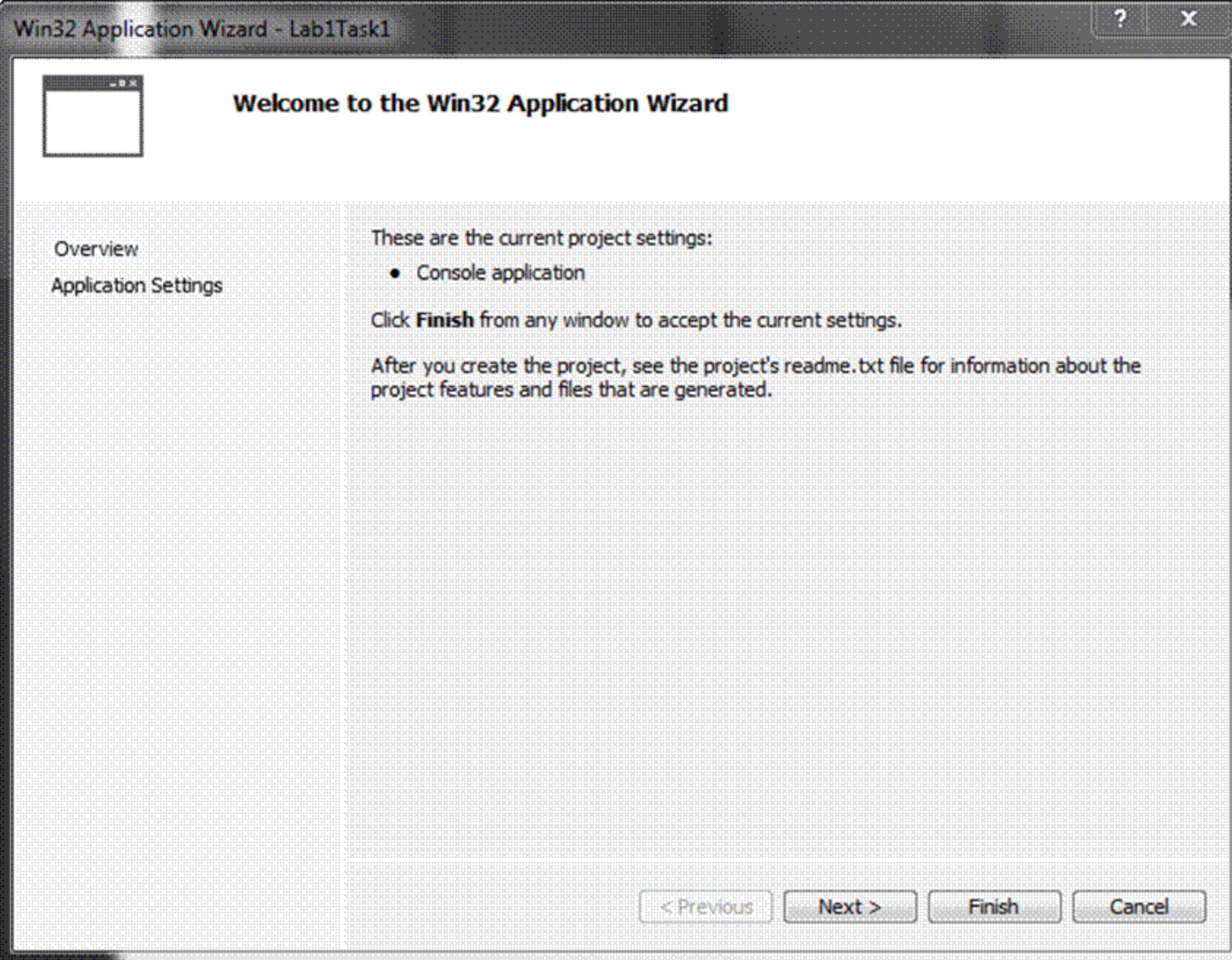




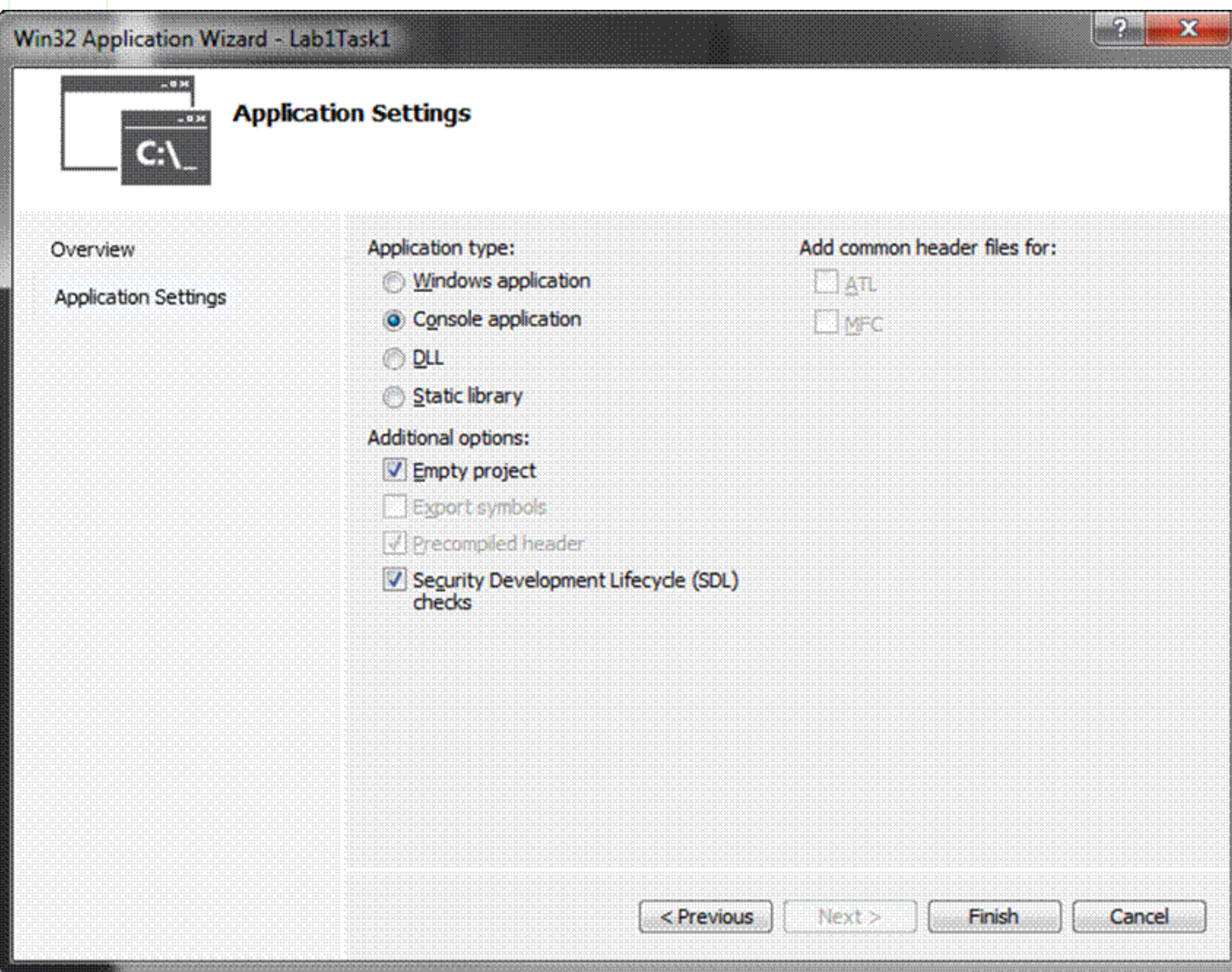
In this window, select Visual C++ Projects on the left and then select Win32 Console Application on the right. (See the gray shaded entries above). Next, give the project a name. A naming scheme will make it easier to find different solutions and projects later. At this time, name this project “Lab1Task1”. You also need to enter the Location; it should be “Z:\cpts121” 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 Application Settings Tab.



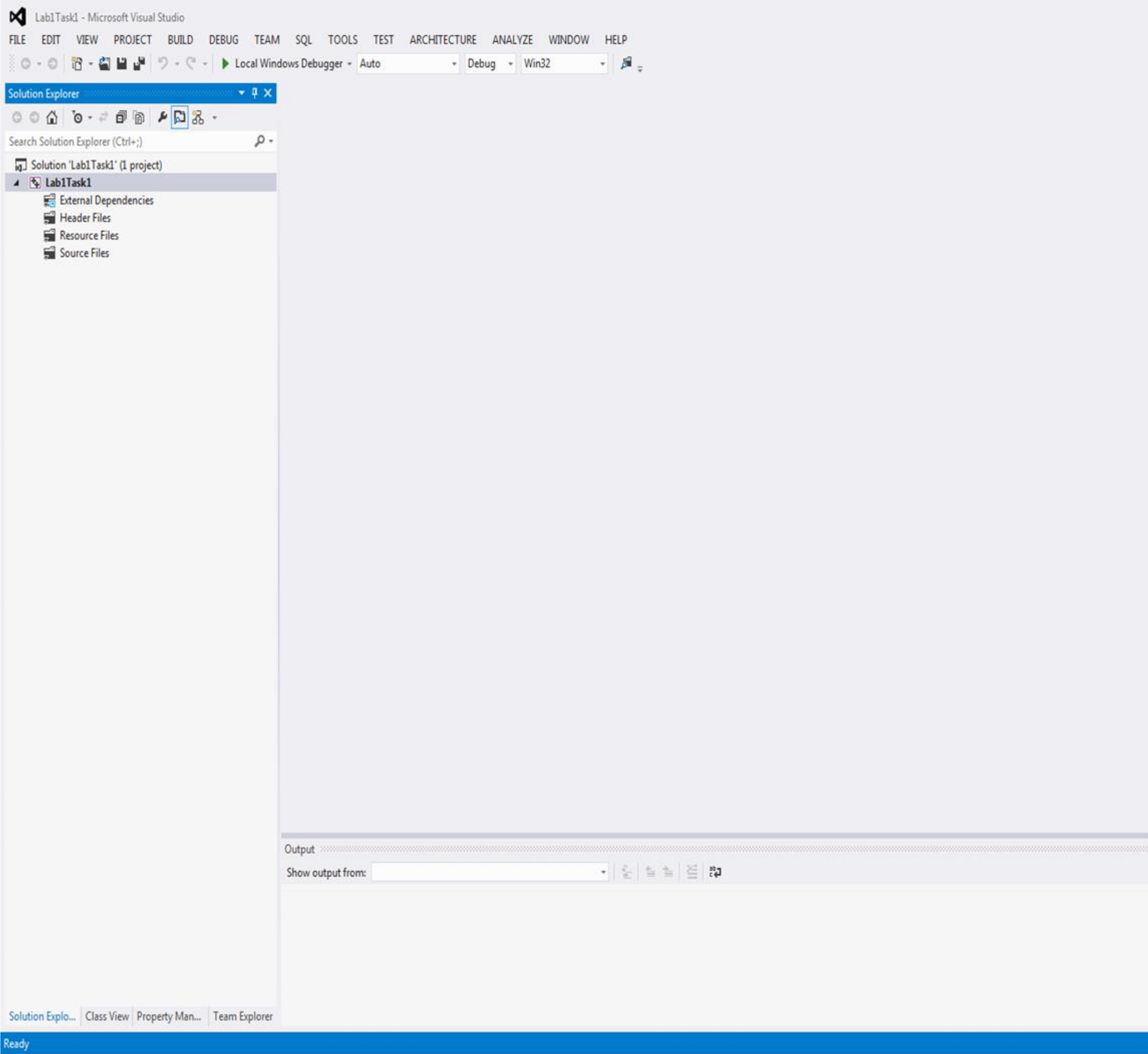


You will see the Wizard window change to provide several choices. Select the Console application (should be the default Application type already) and Empty project as shown in the screen image below. Then click the Finish button.

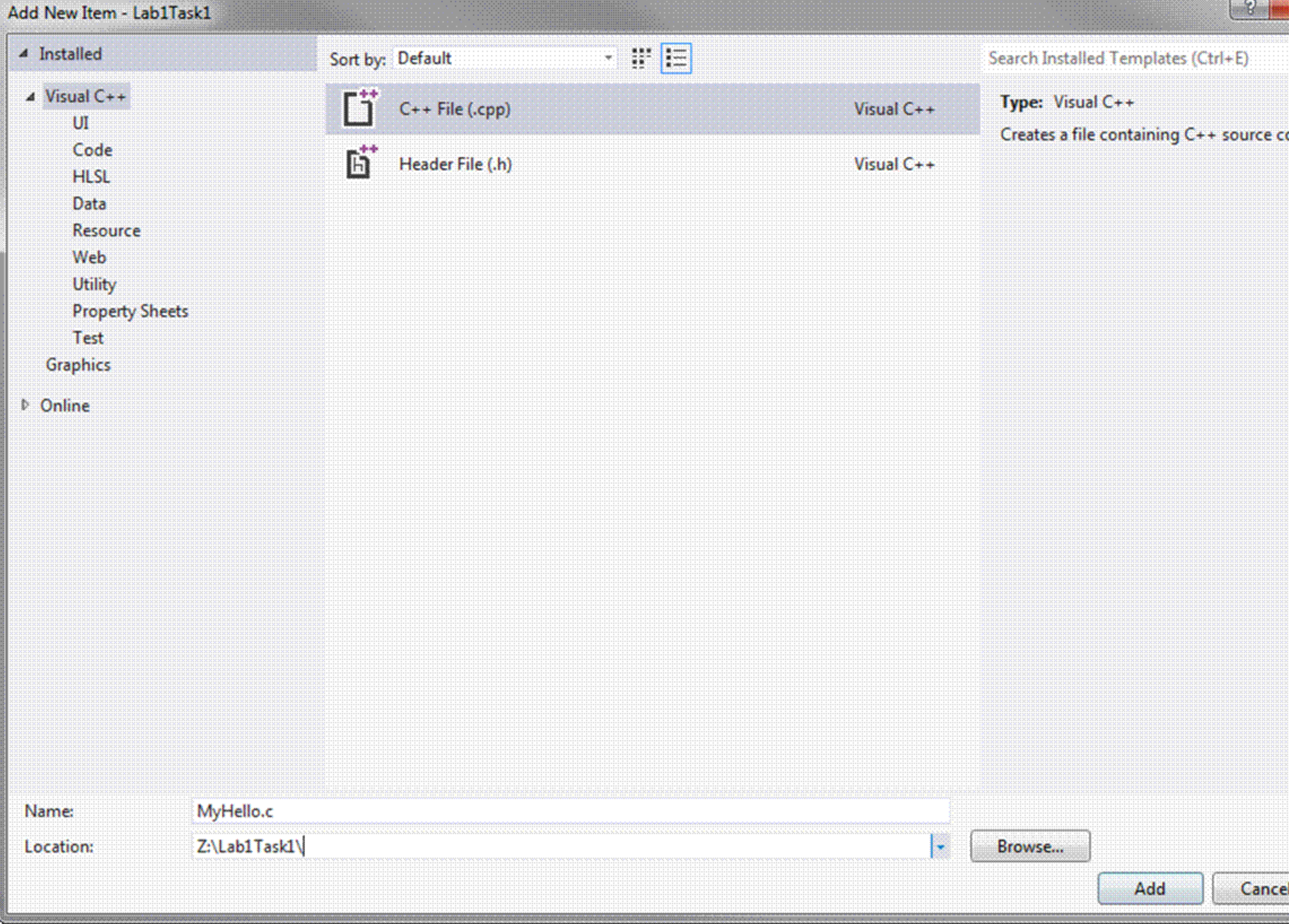


The IDE Window will change a little bit now. There is now information on the left hand side of the window in the Solution Explorer.





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 121, Spring 2014; Lab Section X
* Programming Assignment: Lab1Task1
* Date: January 20, 2014
* Description: This program prints out a simple "hello world" message.
*****/
```

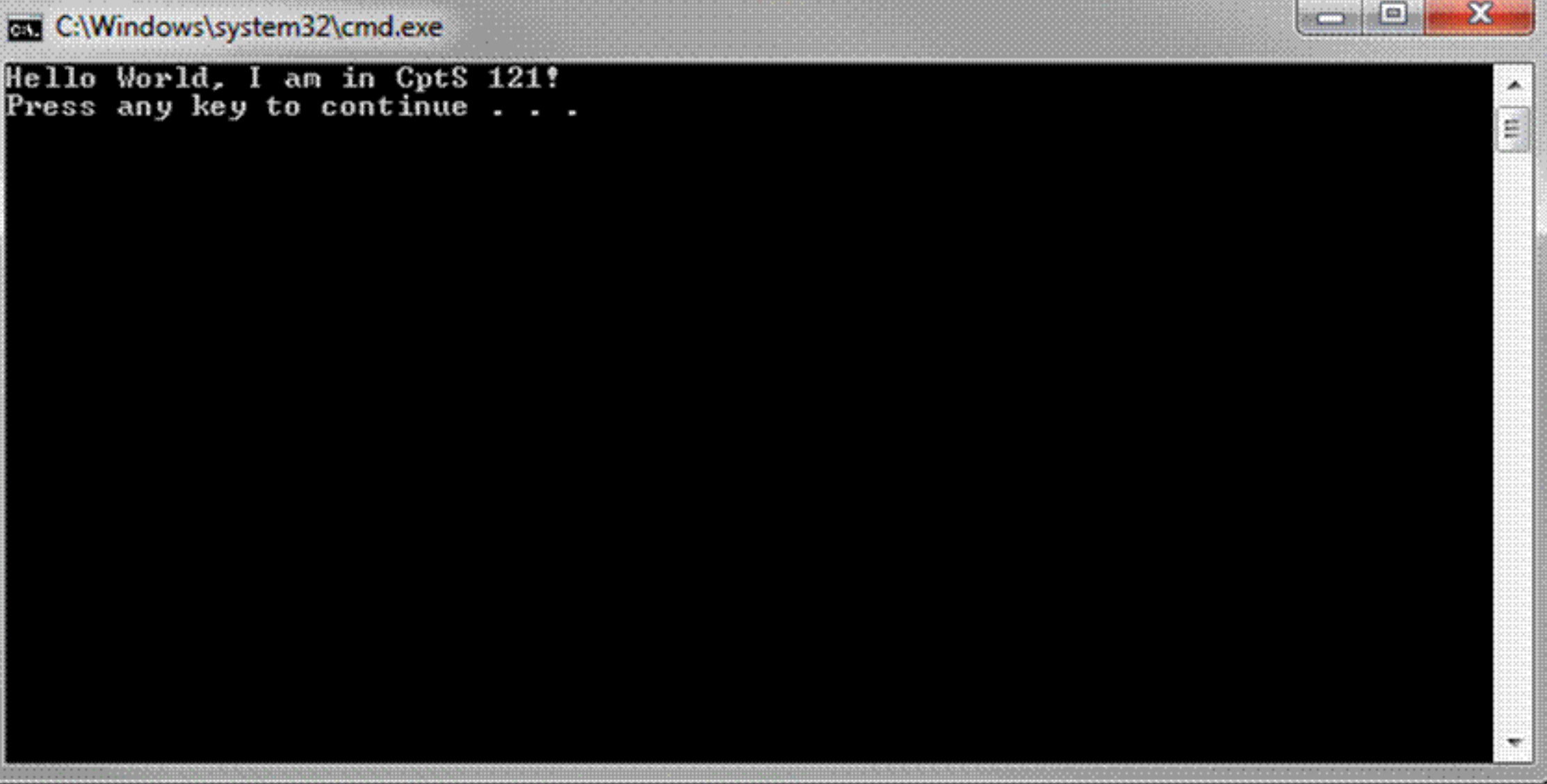
For this first program you will only need one executable C statement to output a message. Your program should output "Hello World, I am in CptS 121!" with a printf ( ) statement.

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.

Life does not get much better than the joy experienced once you see the window below! ☺





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 program that prompts the user for two integer numbers (called number1\_int and number2\_int) and two floating-point numbers (called number1\_float and number2\_float). Perform the following operations on the numbers:
  - Add number1\_int and number2\_int together and print the result as an integer value
  - Subtract number1\_float from number2\_float and print the result as a floating-point value
  - Multiply number1\_int by number1\_float and print the result as an integer value
  - Divide number1\_int by number2\_int and print the result as an integer and a floating-point value (Ask yourself: What happens when you divide and integer by and integer? Does the result change because you print it as a floating-point number?)
  - Divide number1\_int by number2\_float and print the result as an integer and a floating-point value (Ask yourself: How does this compare to the previous operations? We are now dealing with mixed data types.)
  - Explicitly cast number1\_int as a floating-point value and divide by number2\_int. Print the result as a floating-point value.
  - Try to mod number1\_float by number2\_float. Does the program compile? If not, fix it so that it does.
  - Determine if number1\_int and number2\_int are even or odd. Print 0 if even and 1 if odd. You may not use “if” statements. Use the mod operator instead.
- b. Write a program that prompts the user for inputs into Ohm’s Law and determines the voltage. Print the voltage value. Ohm's Law:  $V = IR$ , where V is the voltage, I is the current, and R is the resistance of the circuit (all values are integers).
- c. Write a program that prompts the user for inputs into Joule’s Law and determines the power. Print the power value. Joule's Law:  $P = (V^2) / R$ , where P is the power, V is the voltage, and R is the resistance of the circuit (all values are integers). Do we have loss of precision? What kind of division is being performed?
- d. Write a program that prompts the user for inputs into the third order polynomial equation provided and determines the y value. Print the y value. The third order polynomial equation is:  $y = 3 * ax^3 + (1/4) * bx^2 + 10 * cx + (-5) * d$  (x, y, a, b, c, and d are integer numbers). Be careful here! What is (1 / 4) as an integer result? We do not want integer division in this case! How do we fix it? Do we ultimately have loss of precision?
- e. Write a program that prompts the user for inputs into the circumference of circle equation provided and determines the circumference value. Print the circumference value. Circumference of a circle:  $circumference = 2 * PI * radius$  (circumference and radius are floating-point numbers; PI should be defined as a constant floating-point number and should not be entered by the user).

**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 Sloan 353 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.