Lab 1

Introduction to C Programming

Learning Objectives

After completing this section you will be able to create a new program in C++. Compile the program and execute it.

Basic Lab Instructions!

- Talk to your classmates for help.
- You may want to bring your textbook to future labs to look up syntax and examples.
- Stuck? Confused? Have a question? Ask a TA/Lab Engineer for help, or look at the book or past lecture slides.
- Complete as many problems as you can within the allotted time. You don't need to keep working on these exercises after you leave the lab.
- Before you leave today, make sure to check in with one of the Lab Engineers/TAs in the lab to get credit for your work.

Lab Tasks

Getting Started with Visual Studio

Visual Studio is a sophisticated but easy to use integrated development environment (IDE) for C++ (and may other languages!) You will see that this environment recognizes C++ keywords and puts them into color. This guide assumes that you have access to a machine with Visual Studio 2010 or above installed. You can download the express editions for free form the Microsoft website.

Starting the Visual Studio Environment

1. Click on Start in the task bar, and then move the mouse to All Programs.
2. Choose Microsoft Visual Studio 2010 (or above)

Task #1: Creating a Project in Visual Studio (to compile and run programs)

Even though we wish only to compile and run some fairly small and simple programs, we still need to create a project. In general, this enables a programmer to go back later and open and modify the source code of this project.

Different programs can be contained in different projects. In Visual Studio, a solution can contain more than one project. In the following, let's create a solution called CS110Labs and a project called Lab00. When you want to work on your Lab01 next time, you can open the solution CS110Labs and add another project called Lab01.

To create a solution and a project:

1. Click File in the top menu bar of Visual Studio, and then select the New... option, and select project (Figure 1.1).
2. This will bring up a dialog box that contains many project possibilities (Figure 1.2).
3. Click on "Win32" in the left pane of the window (Figure 1.2).
4. Select "Win32 Console Application" from the right pane of the window (Figure 1.2).
5. Enter a name for the project in the lower part of the windows that says <Enter name> (Figure 1.2): Lab00.
6. Make sure that the location specified for the project is on drive H: If it is not, replace the location with H:.
7. Now enter the name for the solution: CS110Labs.
8. Press OK to create the solution and the project.
9. A Win32 Application Wizard window will appear (Figure 1.3). Select "Application Settings" on the left side of this window.
10. Under additional options, check the "Empty Project" box (Figure 1.4).
11. Select Finish (Figure 1.4).

When we complete this process, Visual Studio creates a folder H:\textCS110Labs. This is the folder where the solution is located. You will see a file with extension .sln here (CS110Labs.sln). This is the file you need to click if you want to reopen your solution next time.

Notice the folder called Lab00 is located in our solution folder. This is the project folder for our Lab00 and the source files (.c and .h) are saved inside this folder. If you have your files elsewhere (e.g your USB pen or My Documents) you should copy them into this folder. When you create your Lab01 project next time, another project folder called Lab01 will be created in the solution folder. In your Visual Studio window you will see a view like this at the top left in the following Figure 1.5. If you do not see this view select View > Solution Explorer or click on the solution explorer button.

When you have created your Lab01 next time, the view will look like (Figure 1.11). Now we see how Visual Studio helps us to manage our projects and files.

Task #2: Add some source code to this project

We have two options when we want to add source code to our project. First is to create new file (Add New Item) and the second is to use an existing item (Add Existing Item). Here we will try out the first option.

1. Right click the project Lab00 in Solution Explorer, and then select the Add > New Item option (Figure 1.6).
2. You should see a window with Installed Templates (Figure 1.7). Select Visual C++ > C++ File (.cpp).
3. Enter MyFirstProgram_v1.c in Name textbox.
4. Press Add button.
5. You should now see the name of this file (MyFirstProgram_v1.c) displayed under the folder Source Files in the upper left panel.
6. Write some code in the source file MyFirstProgram_v1.c (Figure 1.8).

```
/*********************************************************
 * MyFirstProgram_v1 -- program to print out "Welcome to CS110!!". *
 * Not an especially earth-shattering program. *
 * *
 * Author: Anis Rahman. *
 * *
 * Purpose: Demonstration of a simple program. *
 * *
 * Usage: *
 * Runs the program and the message appears. *
 *********************************************************/

#include <stdio.h>
#include <stdlib.h>

int main()
{
  printf( "Welcome to CS110!!" );

  getchar();
  return EXIT_SUCCESS;
}
```

To get the program working we need to do two important things. First we need to compile (and link) the program, then we need to run the program. Here is how to do that.

You wish to compile the source code of this project (MyFirstProgram_v1.cpp):

7. To compile, select the Build menu in the top menu bar (Figure 1.9).
8. Select the option Build Lab00 (assuming that Lab00 is the name of your project).
9. A lower window will show the results of this process. (As it is, MyFirstProgram_v1.c file will compile successfully.) You will see an output like this. You should have 0 error(s).

```
1>------ Build started: Project: Lab00, Configuration: Debug Win32 ------
1>  MyFirstProgram_v1.c
1>  Lab00.vcxproj -> c:\users\anis\documents\visual studio 2010\Projects\CS110Labs\Debug\↩
    Lab00.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

You wish to run the executable file created in the compilation process:

10. To run (execute), select the Debug menu in the top menu bar (Figure 1.11).
11. Select the option Start Without Debugging.

Rebuild Solution will rebuild all the projects in the solution as defined by the configuration manager. If your projects are small this is a convenient option.

12. A black 'DOS-type' window will appear showing the output of your program.

To edit the source code (modify the program):

13. To edit, double click on MyFirstProgram_v1.cpp under the folder Source Files in the upper left-hand panel if it is not already displayed in the main panel. This will load your file into the main panel.
14. Modify the "Welcome to CS110!!" data to read something a little different, say "Greetings human! Welcome to CS110!!"

To recompile the source code of this project:

15. To recompile, select the Build menu in the top menu bar.
16. Select the option Rebuild Lab00.
17. A lower window will show the results of this process.

You wish to rerun the executable file created in the compilation process:

18. To run (execute), select the Debug menu in the top menu bar.
19. Select the option Start Without Debugging.
20. A black 'DOS-type' window appears showing the new output of your program.

Task #3: Reviewing the Visual Studio Environment

Now let's try everything again from scratch!

1. Open notepad, and type in the code listed below.

```
1  /*********************************************************
2   * ArithmeticOperation -- program to print out distance. *
3   * *
4   * Author: Anis Rahman. *
5   * *
6   * Purpose: Demonstration of a simple program. *
7   * *
8   * Usage: *
9   * Runs the program and the computes the distance between two points. *
10  *********************************************************/
11
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <math.h>
15
16 int main()
17 {
```

```
18    // Declare variables
19    double x1, y1, x2, y2,
20      a, b, distance;
21
22    // Initialize variables
23    x1 = 1;
24    y1 = 5;
25    x2 = 4;
26    y2 = 7;
27
28    // Compute the sides of the triangle
29    a = x2 - x1;
30    b = y2 - y1;
31
32    // Compute the distance
33    distance = sqrt( a*a + b*b );
34
35    // Print the distance
36    printf( "The distance between the two points is %.3f", distance );
37
38    // Wait for a keypress before exit
39    getchar();
40
41    // Exit the program
42    return EXIT_SUCCESS;
43  }
```

2. Save the file named as ArithmeticOperation.c in the same directory as MyFirstProgram_v1.c.

3. Click Project in the top menu bar of Visual Studio, and then select the Add Existing Item option.

4. Since we have already copied the files to the right place you will see the icon of the file you need to add. Select ArithmeticOperation.c. If you wish, by selecting the arrow beside the Look in address box, you can navigate to the folder in which you saved the ArithmeticOperation.c file. To see the C++ files in this folder, set the Files of type (see bottom box of Look in menu) box to contain Visual C++ Files (which may already be the default value).

5. You should now see the name of this file (ArithmeticOperation.c) displayed under the folder Source Files in the upper left panel.

6. Double click on ArithmeticOperation.c under "Source Files" on the left-hand side to display the contents of the file in the main portion of the Visual Studio development environment.

7. Compile the program and execute it.

If you get an error saying

```
1  1>------ Build started: Project: Lab01, Configuration: Debug Win32 ------
2  1>  ArithmeticOperation.c
3  1>MyFirstProgram_v1.obj : error LNK2005: _main already defined in ArithmeticOperation.obj
4  1>c:\users\anis\documents\visual studio 2010\Projects\CS110Labs\Debug\Lab01.exe : fatal ↩
       error LNK1169: one or more multiply defined symbols found
5  ========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ==========
```

you have more than one source file containing a main function. Remove all except the one you want. Here, we need to remove MyFirstProgram_v1.c.

1. Right click on the MyFirstProgram_v1.c and select remove. Removing does not mean that you are deleting the file. The file is just removed from the project.
2. Compile the program and execute it.

Task #4: Optional

If you modify a file with a deliberate error, you can see what sort of message this error generates when you try to recompile.

- try omitting a semicolon from the end of a statement (and then recompiling).
- try a different integer name

Hand in

Hand in the source code from this lab at the appropriate location on the blackboard system at LMS. You should hand in a single compressed/archived file that contains the following.

1. An plain text file named OUTPUT.txt that includes a) author information at the beginning, b) a brief explanation of the lab, and c) any comments, or suggestions.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.
2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.
3. In-class lab time is not intended as free time for working on your program assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the program assignment.

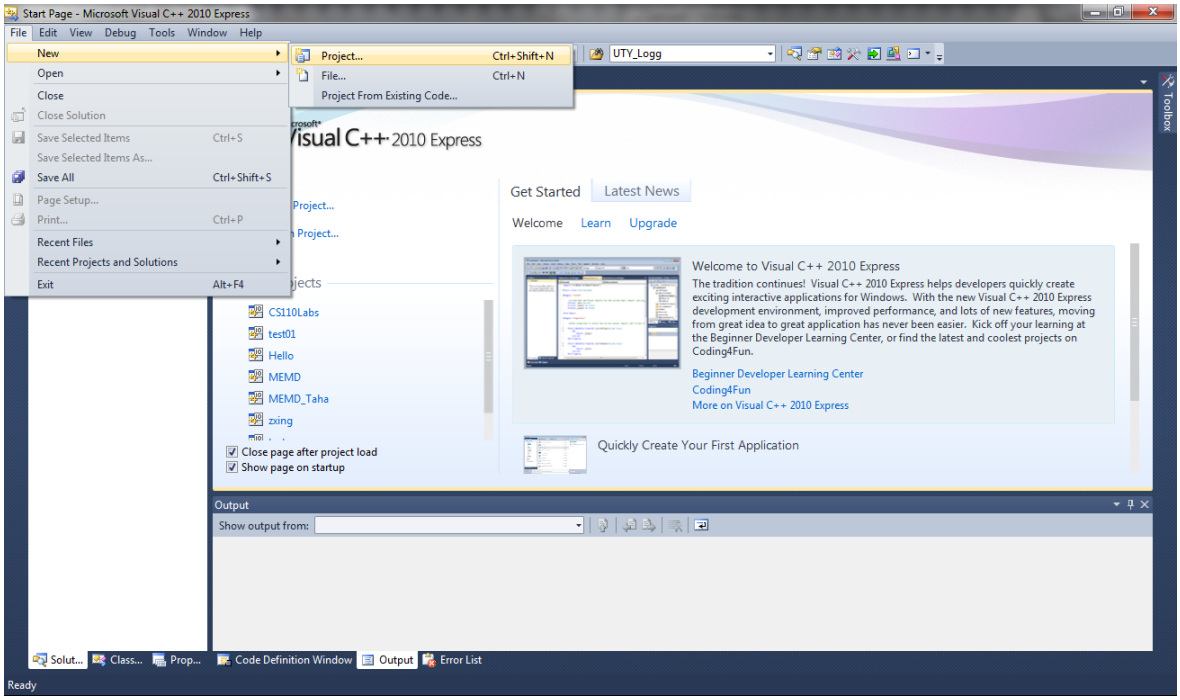

Figure 1.1: Screenshot of new project

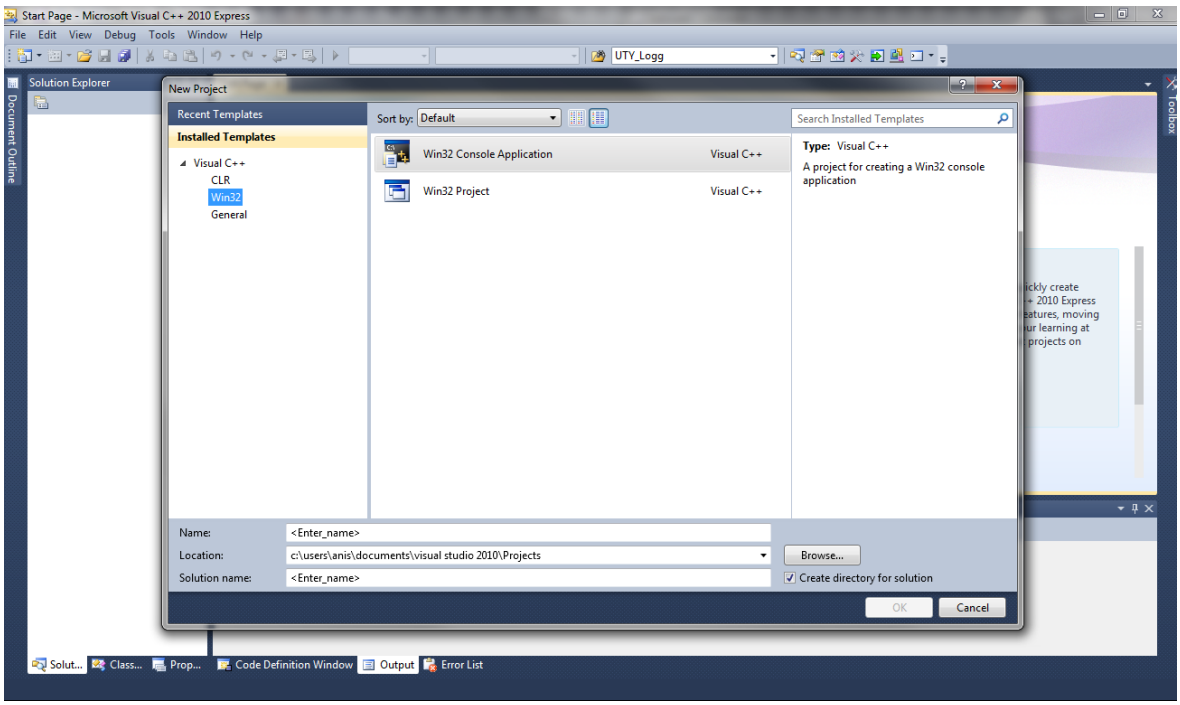

Figure 1.2: Screenshot of project types

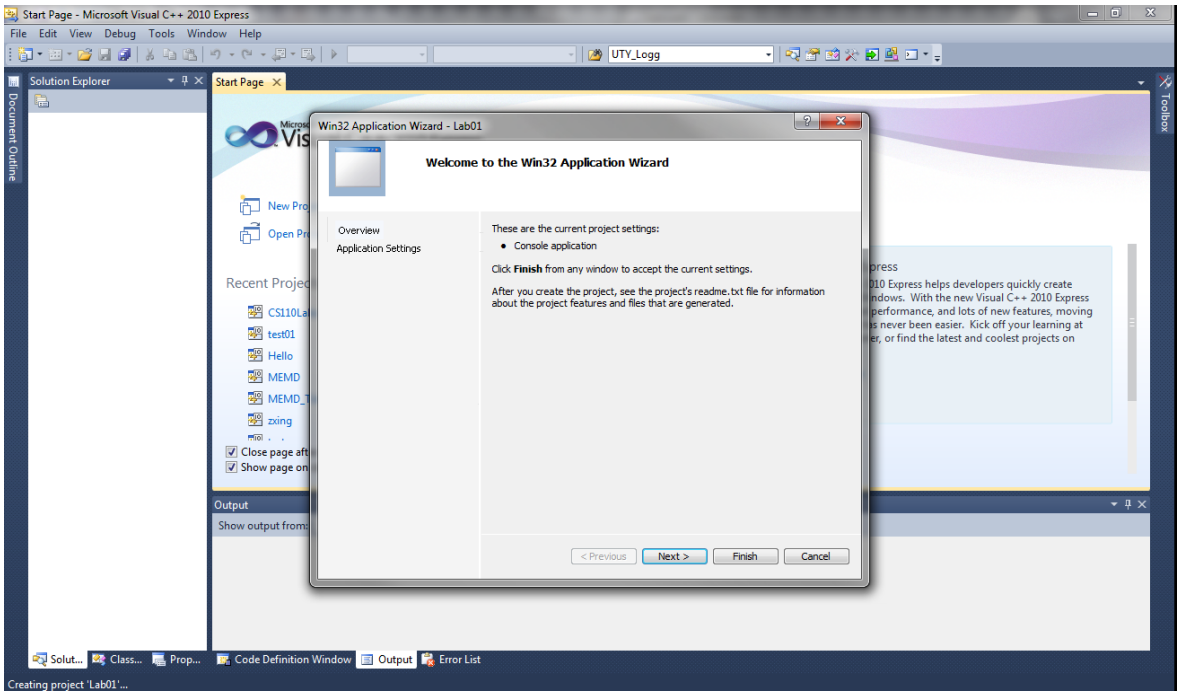

Figure 1.3: Screenshot of Win32 console project

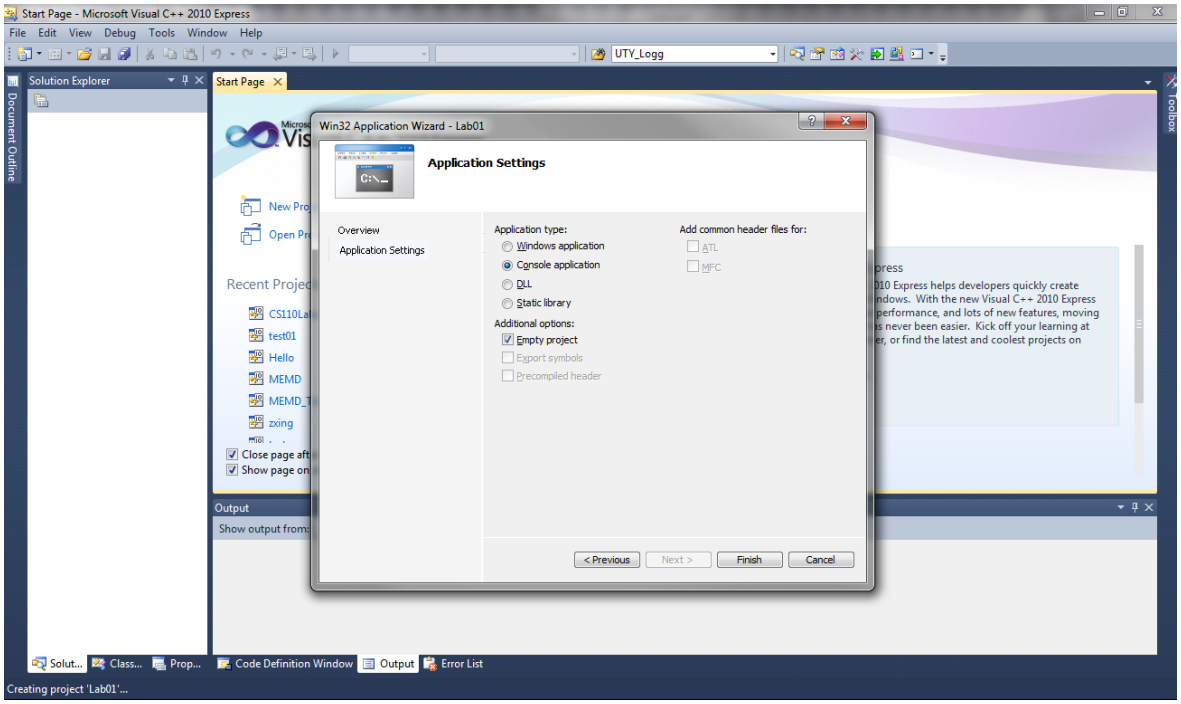

Figure 1.4: Screenshot of application settings window

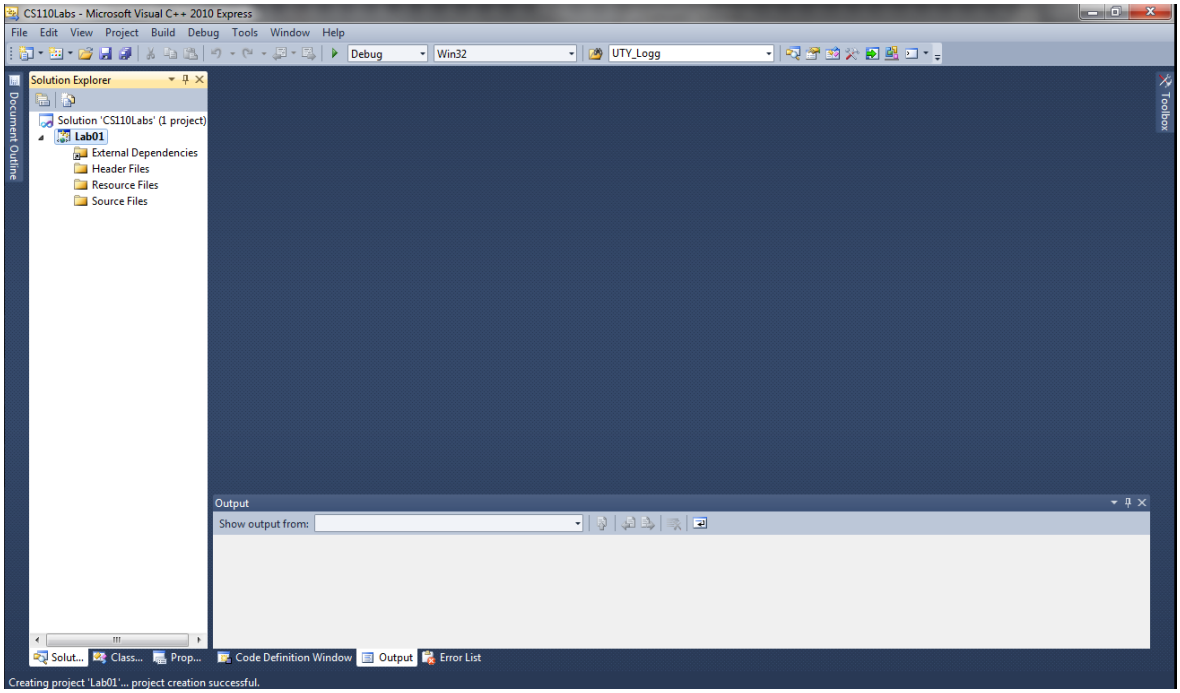

Figure 1.5: Screenshot of empty project

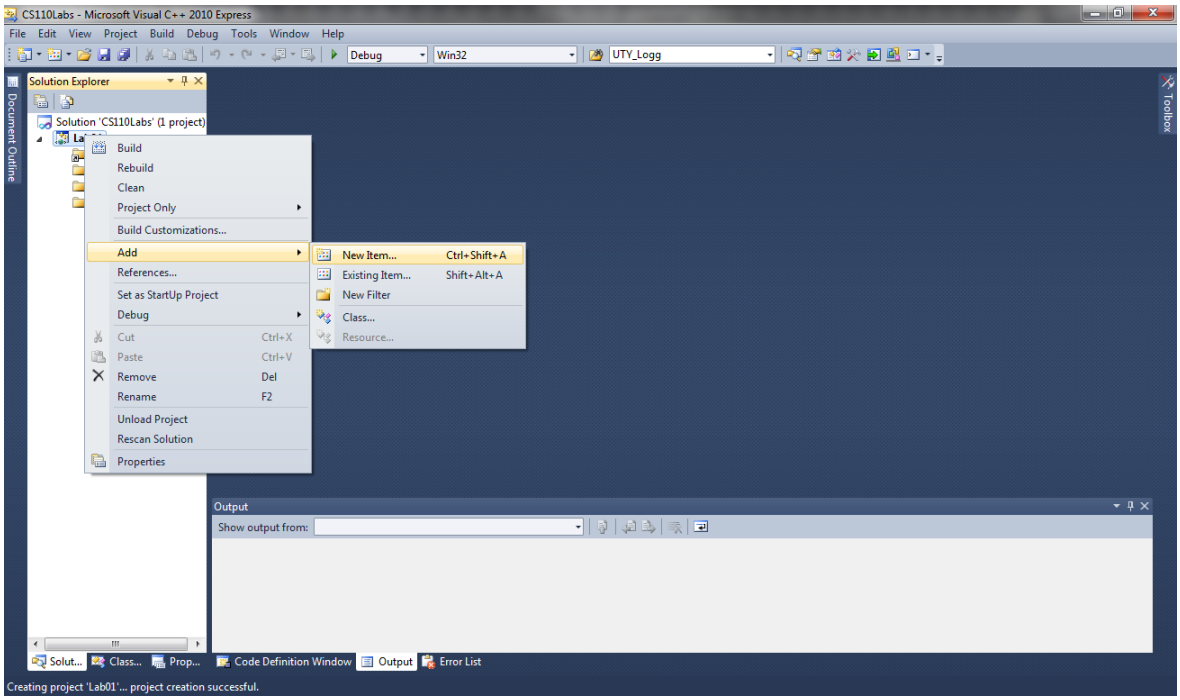

Figure 1.6: Screenshot of adding new item

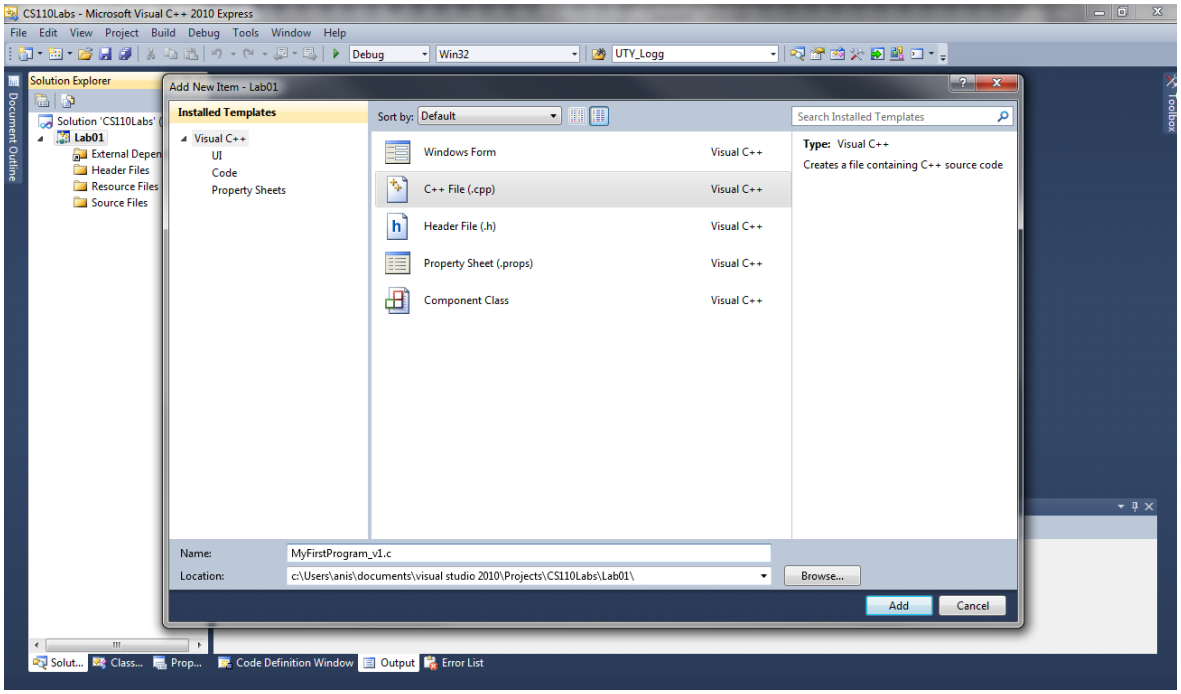

Figure 1.7: Screenshot of available items

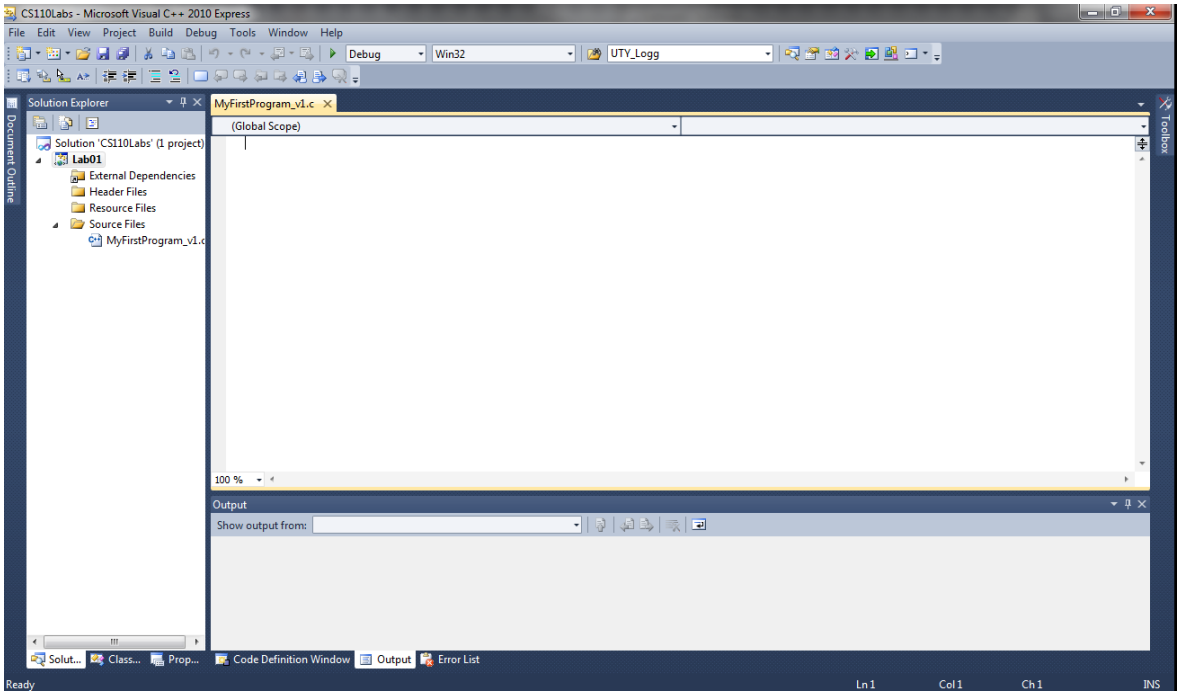

Figure 1.8: Screenshot after adding a new item

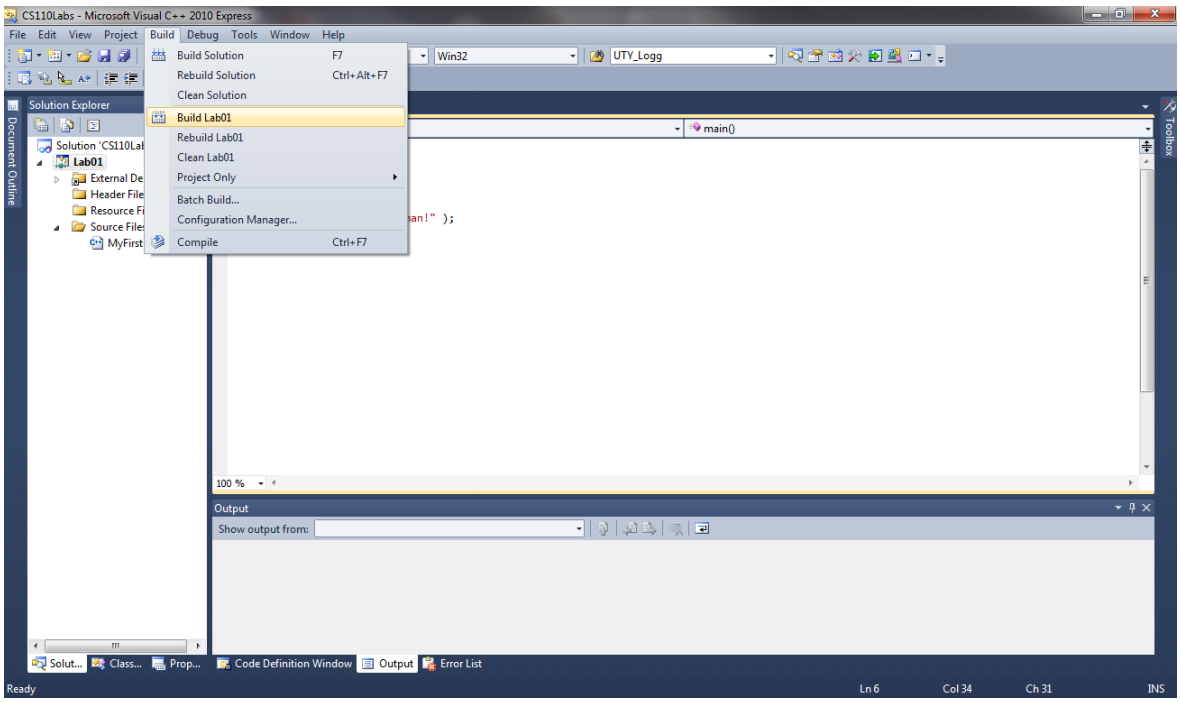

Figure 1.9: Screenshot of building project files

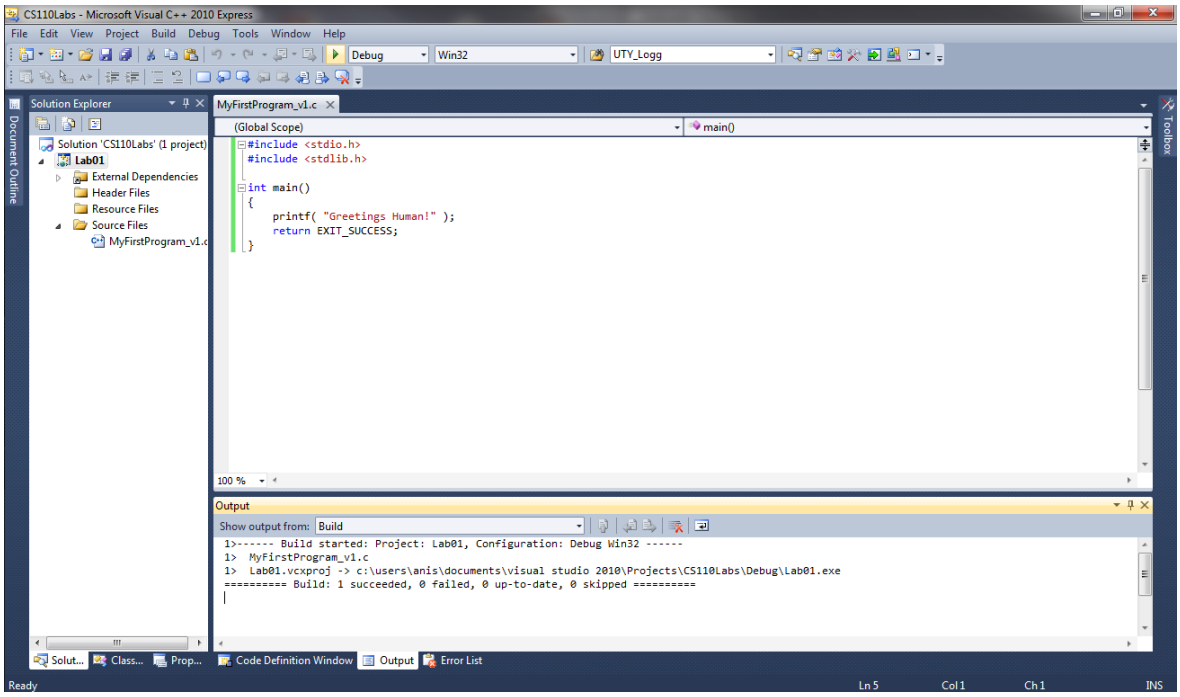

Figure 1.10: Screenshot after successful compilation

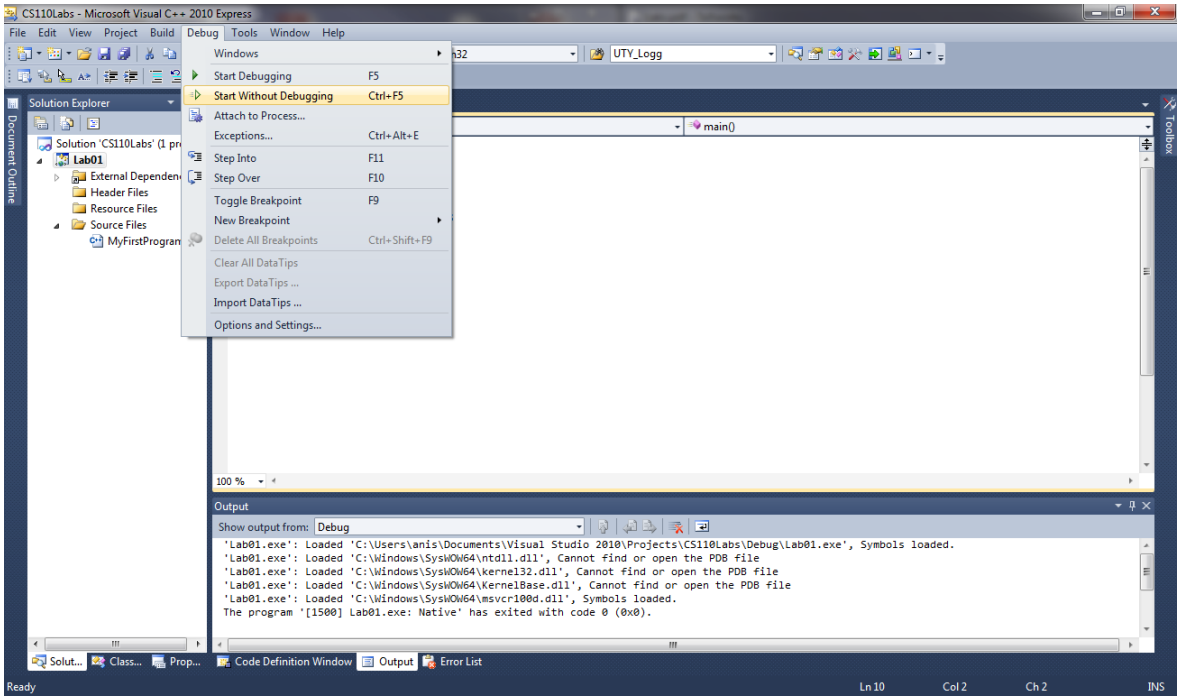

Figure 1.11: Screenshot of solution with multiple projects