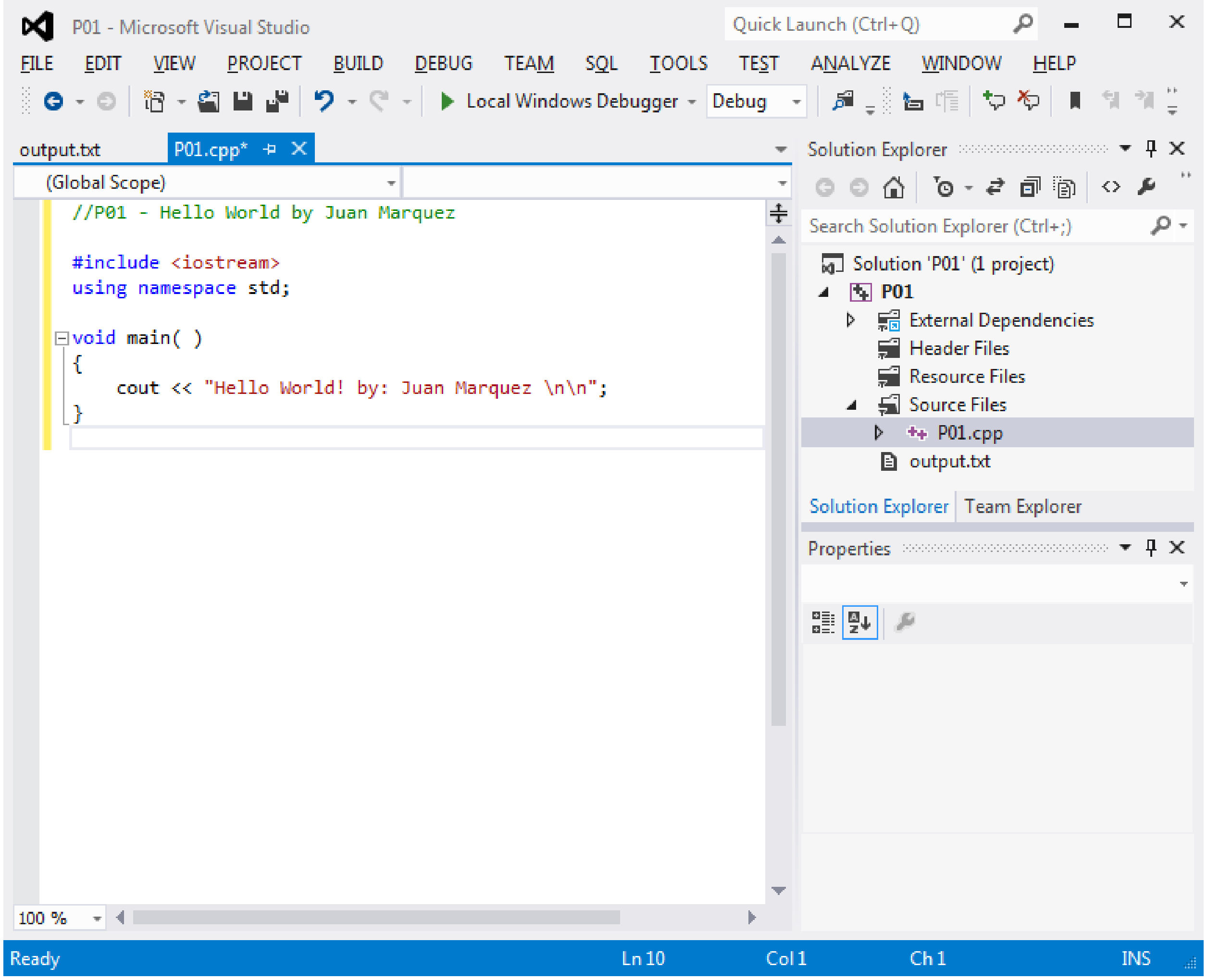
MCC - CIS 162AB - C++ Level I
P01 Hello - 10 points

Introduction to Using Microsoft Visual Studio

All files related to a program are stored in a Project. This document will walk you through the process of creating a Project, and adding a Source file, and text file to it. You will need a separate Project for each assignment, so pay attention to the detailed steps as you go through this document, because you will be doing this throughout the semester. Only the required options and features to complete this assignment are included. Additional details will be revealed in the textbook and lecture notes.

1. The following instructions will lead to the development of the following program:



```
//P01 - Hello World by Juan Marquez

#include <iostream>
using namespace std;

void main( )
{
    cout << "Hello World! by: Juan Marquez \n\n";
}
```

2. Put your name on your storage device (flash disk).
   Consider creating a folder with your name and a document with contact information.

3. Insert storage device into the appropriate connection on the computer.

4. Start Microsoft Visual Studio 2012 **or** Visual Studio Express 2012 for Desktop.

    a. In Mesa Community College's computer lab open **Microsoft Visual Studio 2012**.

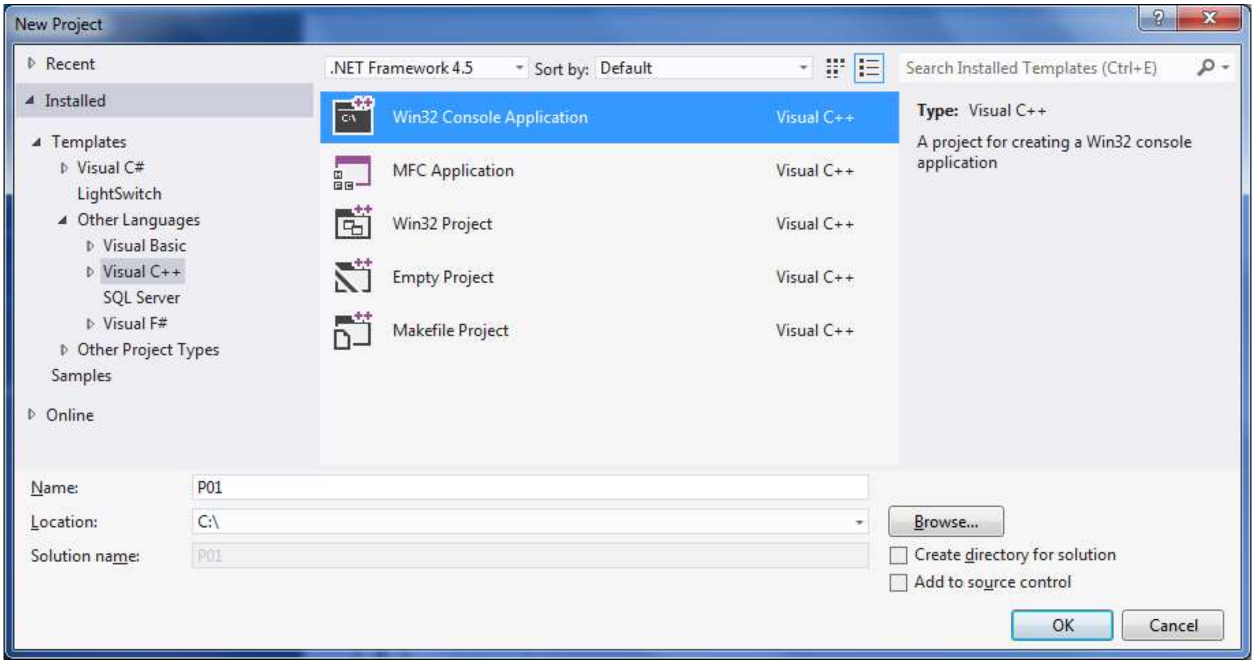
    b. The program to open on your home computer will depend on which version was installed, **Microsoft Visual Studio 2012** or **Visual Studio Express for Desktop.**

    c. If necessary, maximize the application window to fill the whole screen by clicking the **maximize** button in the upper right-hand corner of the window.

    d. The appearance of Start Pages will vary by version and will depend on who used it last and on the options selected.

5. Each assignment must be stored in a separate project.

    a. Click on **File** on the menu bar, scroll down and mouse over **New**, and then slide over and select **Project....**



.NET Framework 4.5 Sort by: Default

Win32 Console Application	Visual C++
MFC Application	Visual C++
Win32 Project	Visual C++
Empty Project	Visual C++
Makefile Project	Visual C++

Type: Visual C++

A project for creating a Win32 console application

Name: P01

Location: C:\

Solution name: P01

□ Create directory for solution

□ Add to source control

    b. The list of Project Types and Templates will vary based on the software version being used (Professional or Express).

    c. From the list of Install Templates, click on **Visual C++**.

    d. From the list of Frameworks on the upper right side select **.Net Framework 4.5** in the dropdown list. If a newer version is installed, use it instead. The latest version is usually selected by default.

    Express Note: Framework option not presented.

    e. The Templates in the middle can be displayed either in medium or small icons. **Select Medium Icons** by clicking on the Medium Icon button which is the button on the far right next to the Sort By box.

f. In the list of Templates scroll down (if necessary) and click on **Win32 Console Application**. **Note:** Selecting anything else will cause your program not to work as described in this document.

g. Enter **Name:** P01 (0 is the number zero and not the letter O)

h. To set the **Location:** use the **Browse...** button to navigate to the drive assigned to your storage device.

i. The **Solution Name** should be the same as the application Name (P01). It may already be grayed out if Create a directory for solution does not have a check mark as instructed on the next step.

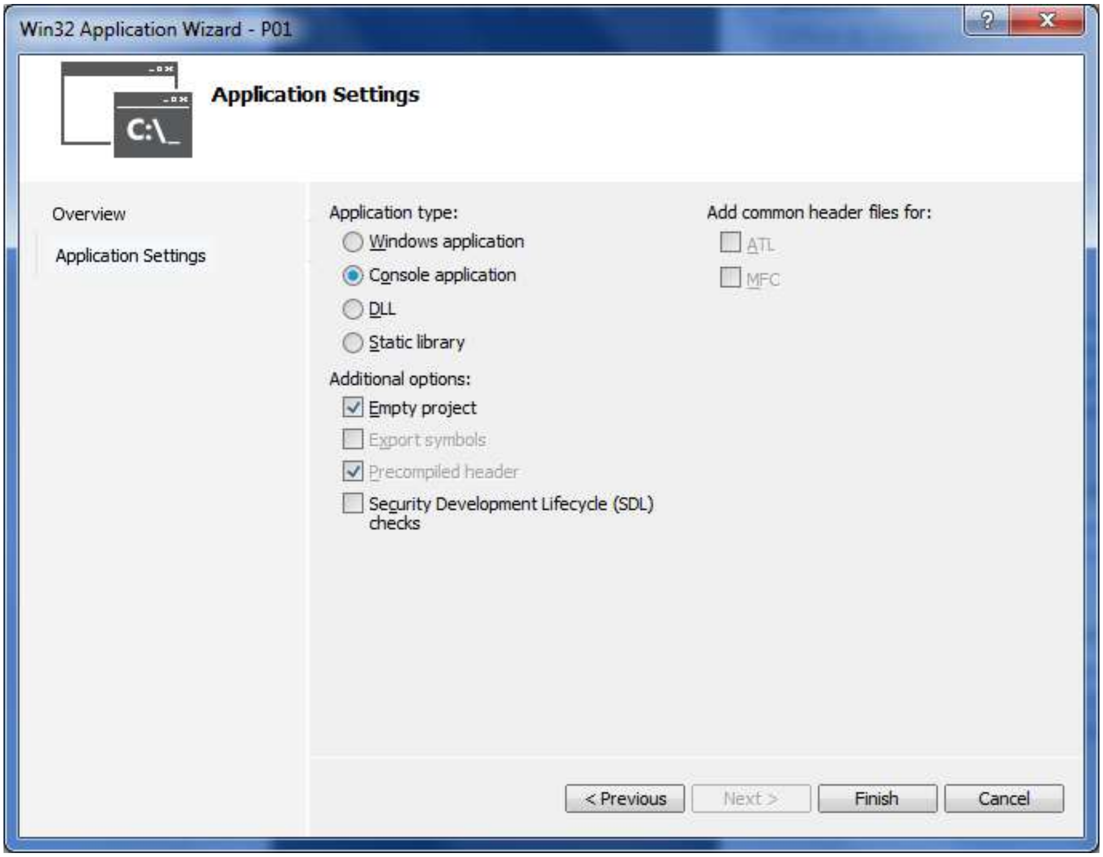
j. Click on the box in front of **Create a directory for solution** to remove the check mark if present.

k. **Do NOT** check the option to add to Source Control.

l. Click on **OK**.

6. After clicking on OK, the Win32 Application Wizard will appear.

a. Click on **Application Settings** on the left size of the window.



Win32 Application Wizard - P01

Application Settings

Overview

Application Settings

Application type:

○ Windows application

● Console application

○ DLL

○ Static library

Additional options:

☑ Empty project

□ Export symbols

☑ Precompiled header

□ Security Development Lifecycle (SDL) checks

Add common header files for:

□ ATL

□ MFC

< Previous | Next > | Finish | Cancel

b. Verify that the radio button before **Console application** is selected.

c. Click on the box in front of **Empty project** to select it.

Always select **Console application** and **Empty project.**
Selecting anything else will cause your programs not to work as described in this document.

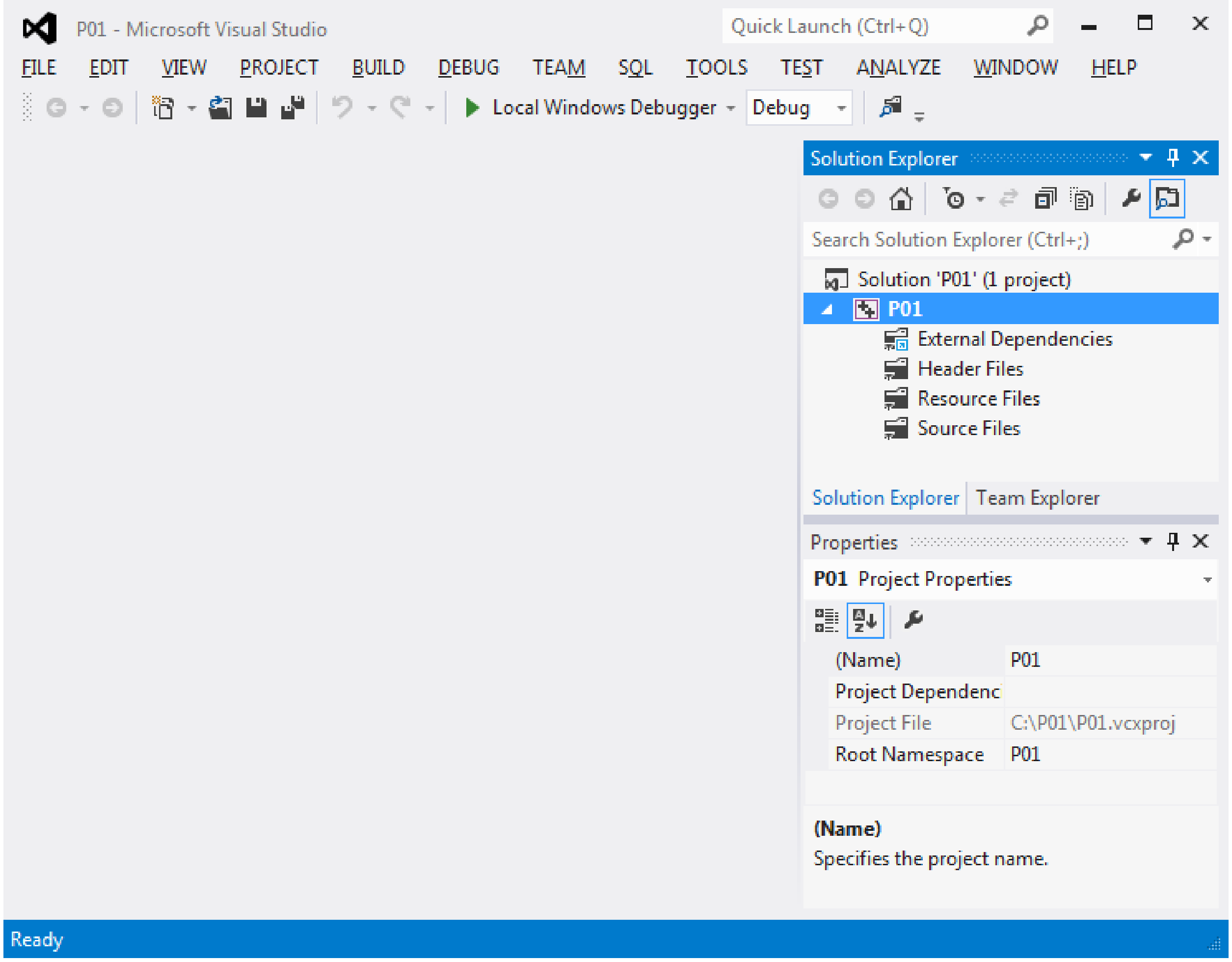
d. Click on the box in front of Security Development Lifecycle (SDL) checks to deselect it.

e. Click on **Finish** to create the project.

7. The new project should appear in the **Solution Explorer** window.

   a. If your workspace window does not display the Solution Explorer, click on **View** on the menu bar and select **Solution Explorer**.

   b. If the ToolBox is opened on the left side, click on the X to close it.

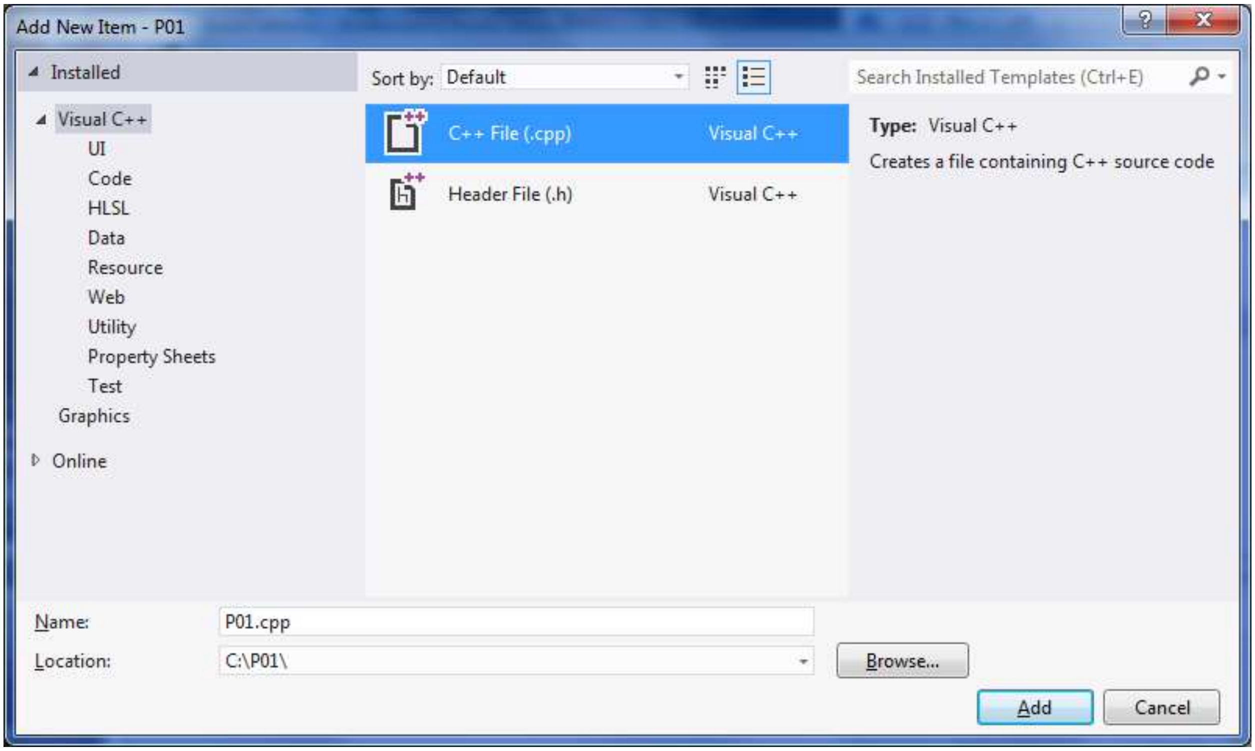


(Name)	P01
Project Dependenc	
Project File	C:\P01\P01.vcxproj
Root Namespace	P01

(Name)
Specifies the project name.

8. The next item that must be **created is a C++ Source File.**

   a. Click on **Project** on the menu bar, and then scroll down and select **Add New Item...**.



Type: Visual C++

Creates a file containing C++ source code

Name: P01.cpp

Location: C:\P01\

   b. On the left side under Install Templates, **Visual C++** should be selected.

   c. In the middle column, scroll up or down and click on **C++ File (.cpp)**.

   d. Enter file **Name: P01**
   The extension .cpp will automatically be added to the source file name.
   Every project will require a C++ source file.

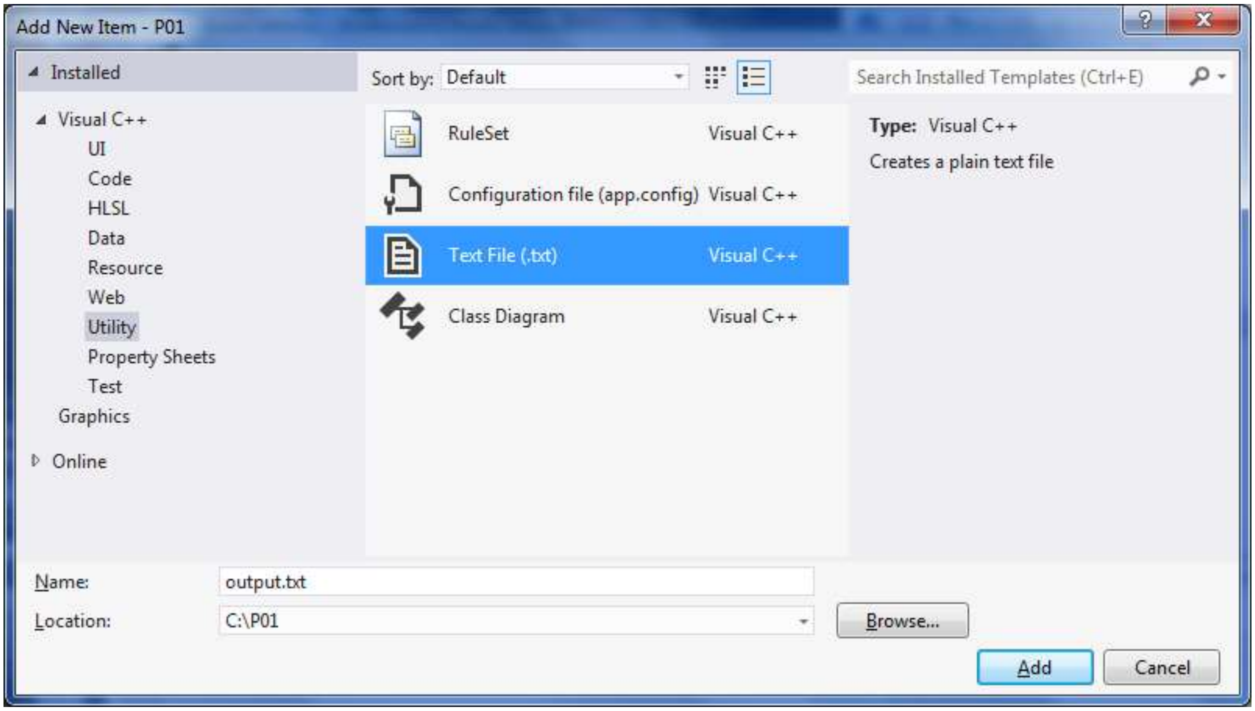
   e. The value for **Location:** will be automatically set based on the values entered when the project was created. It should be going into the P01 folder.

   f. Click on **Add**.

   g. The P01.cpp file should now be listed in the Solution Explorer under Source Files and opened in the workspace window.

9. The last item that will be **created is a Text File** for the sample output.
   This options presented in the middle panel will vary based on the software version being used.

   a. Click on **Project** on the menu bar, and select **Add New Item...**



Type: Visual C++
Creates a plain text file

Name: output.txt
Location: C:\P01

   c. On the left side under Installed Templates, click on **Utility**.

   d. In the middle column, scroll up or down and click on **Text File (.txt)**.

   e. Enter file **Name: output**
      The extension .txt will automatically be added to the file name.
      Just about every project will require a text file for the sample output.

   f. The value for **Location:** will be automatically set based on the values entered when the project was created. It should go into the P01 folder.

   g. Click on **Add**.

   h. The output.txt file should now be listed in the Solution Explorer and opened in the workspace window. The sample output produced by the program will be copied and pasted into output.txt in a later step.

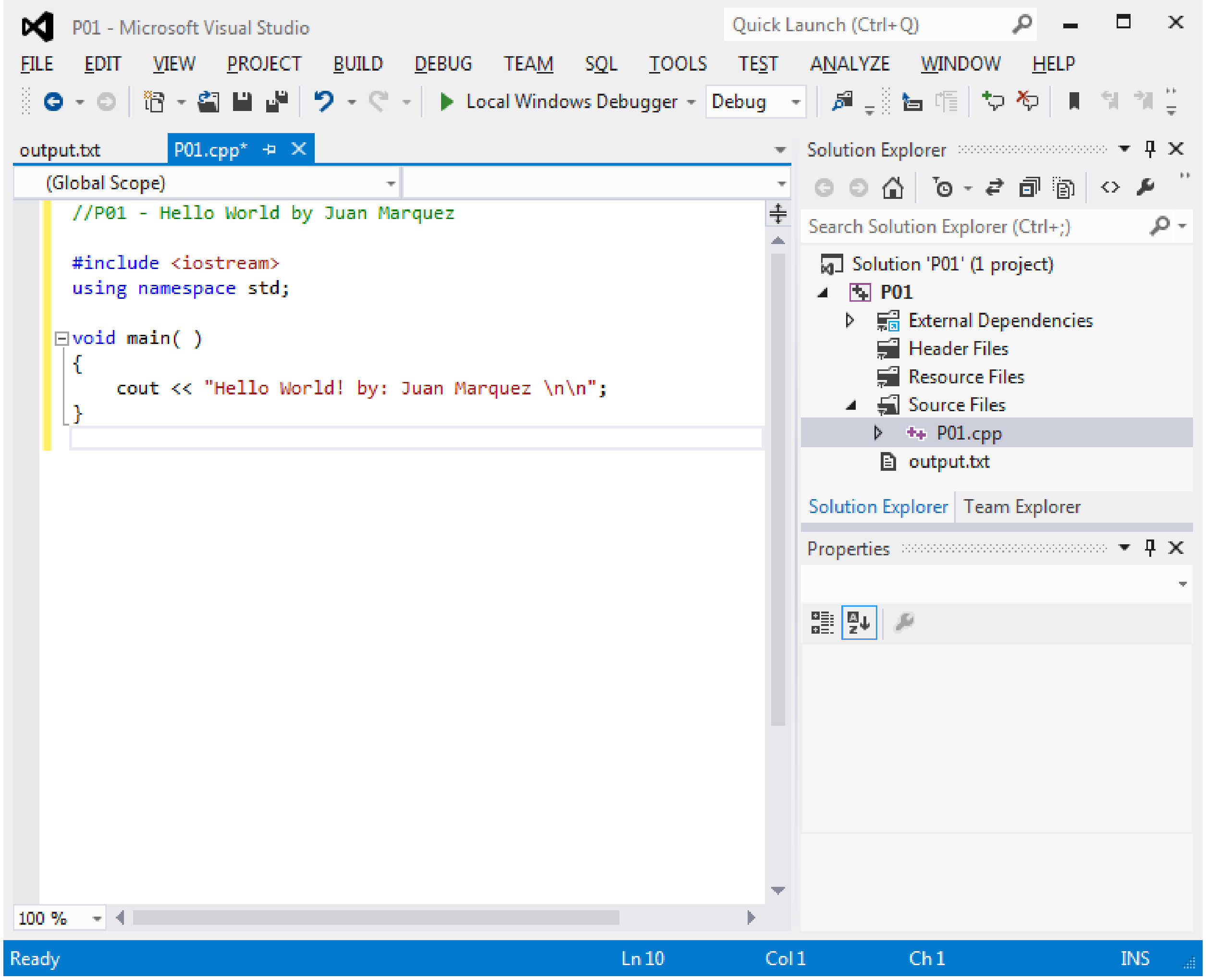
10. Enter the **source code** for the program into the newly created C++ source file.

    a. The file P01.cpp should now be listed under the Solution Explorer within the Source Files folder. If P01.cpp is not visible, click on the **white triangle** in front of the **Source Files** folder name to expand the folder and see the source file P01.cpp.

    b. **Double click** on the file **P01.cpp** to make it the active file for the editor.
    Or click on the tab labeled P01.cpp in the workspace to bring it to the foreground.

    c. **Enter** the Hello World program in the editing window as shown below.
    The editor works more-or-less like a word processor.
    Key words are set to different colors, and it auto-indents on the open and close braces { }.
    Press enter at the end of each line.
    Be sure to change both occurrence of the instructor's name to **your name.**

    d. Click the **Save All** button or select Save All from the File menu. You should save your work at various times during your work sessions to prevent the loss of data.

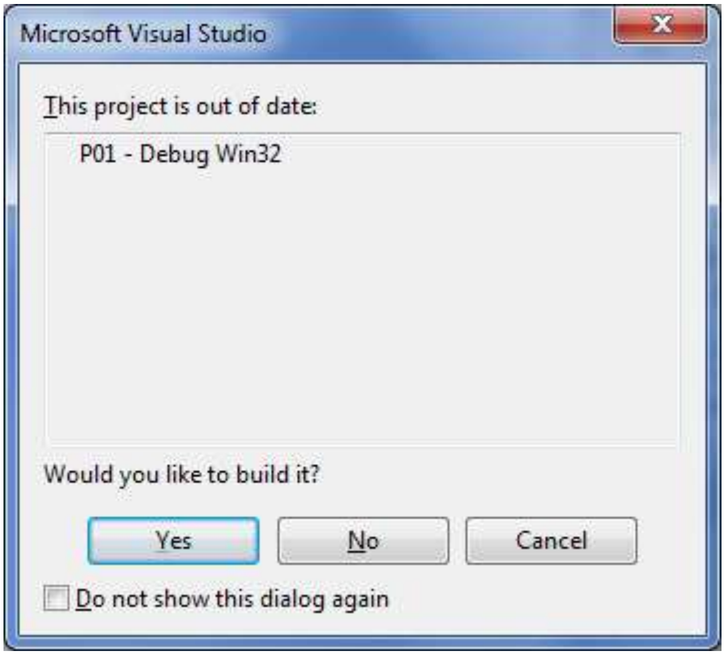


```
//P01 - Hello World by Juan Marquez

#include <iostream>
using namespace std;

void main( )
{
    cout << "Hello World! by: Juan Marquez \n\n";
}
```

11. When you have completed entering or editing a program, it must be **compiled, linked, and executed**. The three steps can be initiated by requesting that the program be executed.

   a. Select **Start Without Debugging** from the **Debug menu**.

   b. Each time the code is changed, the compiler will ask you if you want to rebuild the project. Always click on **Yes**. To skip this step, you can click on the box in front of Do not show this dialog again.



Microsoft Visual Studio

This project is out of date:

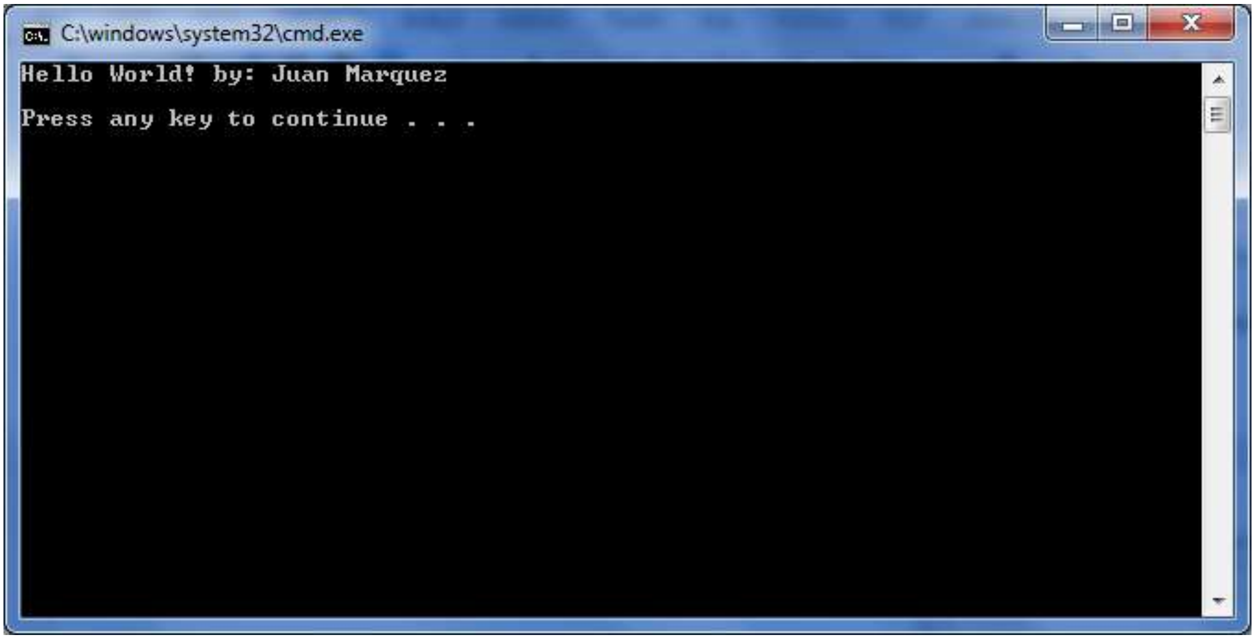
P01 - Debug Win32

Would you like to build it?

   c. The compiler will display informational messages in the Output pane, and a list of errors (if any) in the Error List pane. If your program has syntax errors identified by the compiler, double click on an error to highlight the error message and to move to the line of code where the error occurred. Correct all the errors in the editing window and recompile by selecting Start Without Debugging from the Debug menu again. Keep editing and recompiling until all errors have been corrected. The program must entered as displayed above. The only thing change is the instructor's name to your name. The quotation marks and semicolons are required.

   d. The results from the linker, including error messages, will appear in the Error List pane. If your program has linking errors, you may need some assistance in resolving the problem. Typical errors include using the wrong #include statements, incorrect function names and parameters, or selecting a project type other than "Win32 Console Application". If you suspect a different project type was selected, it may be best to start over and name the second project attempt as P01a.

e. If the program compiles and links successfully, it will be **executed**, and output will appear in a command prompt window like the one shown below:



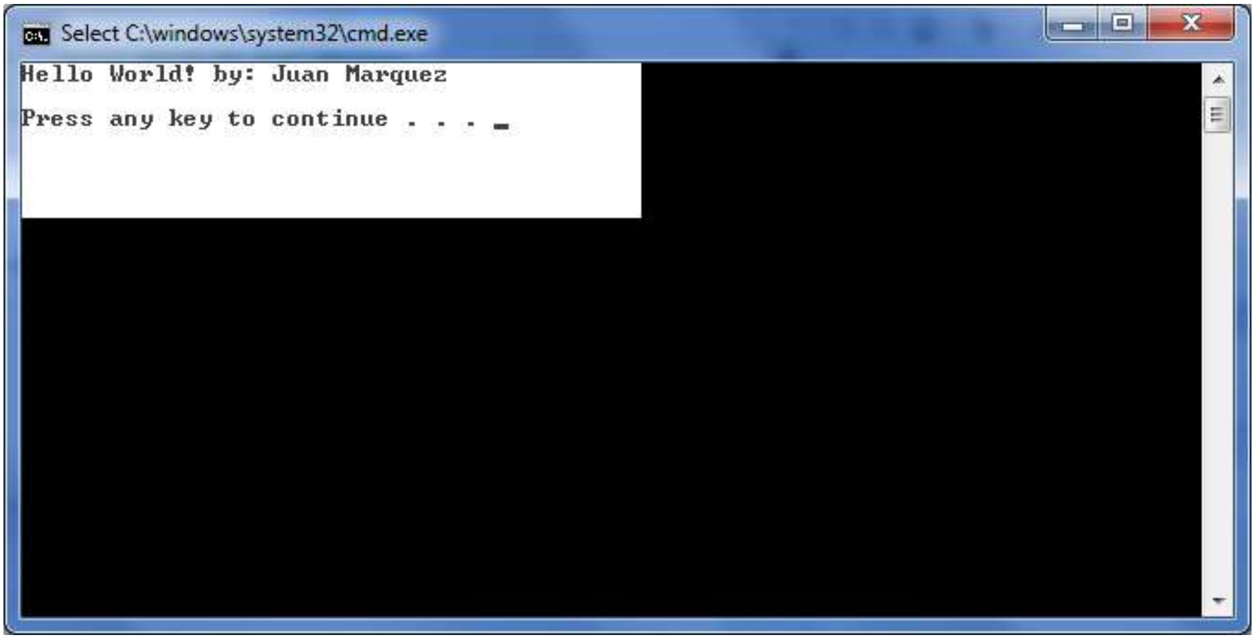
```
Hello World! by: Juan Marquez

Press any key to continue . . .
```

f. If execution does not pause so that the output can be reviewed, be sure to select Start Without Debugging from the Debug menu.

After a program actually produces output, you begin the logic testing. If it works as expected, you're done. Otherwise, you'll need to review the program's logic and revise the program as needed in the editor window. Then you'll need to save, compile, link and execute it again to test the new logic.

12. If the program works as required, capture the sample output.

   a. **Right-click** on output window's title bar.
      On the pop-up menu, scroll down to **Edit**, and then over to select **Mark**.
      This will allow you to select text in the window.
   b. **Click and drag** over the text you need to copy. As you drag the background will turn white.



```
Hello World! by: Juan Marquez

Press any key to continue . . . _
```

   c. Press the **enter key** to copy the selection.
   d. Press the **enter key** again to close the console window and return to Visual Studio.
   e. In the Solution Explore pane, **Double-click on output.txt** to open the file in the editing pane.
   f. **Paste** the sample output into the output.txt file.
   g. Click the **Save All** button.

13. Practice working through syntax errors using the Error List.

   a. **Double-click on P01.cpp** listed under the Solution Explorer to open the file in the editing pane.
   b. **Remove** the semicolon (;) at the end of the cout statement.
   c. Select **Start Without Debugging** from the **Debug menu**.
   d. Click on **Yes** if ask if you want to rebuild the project.
   e. Click on **No** if asked if you want to run the last successful build. Always click No because you want to run the program with your latest changes.

f. One error should be reported on the Error List:

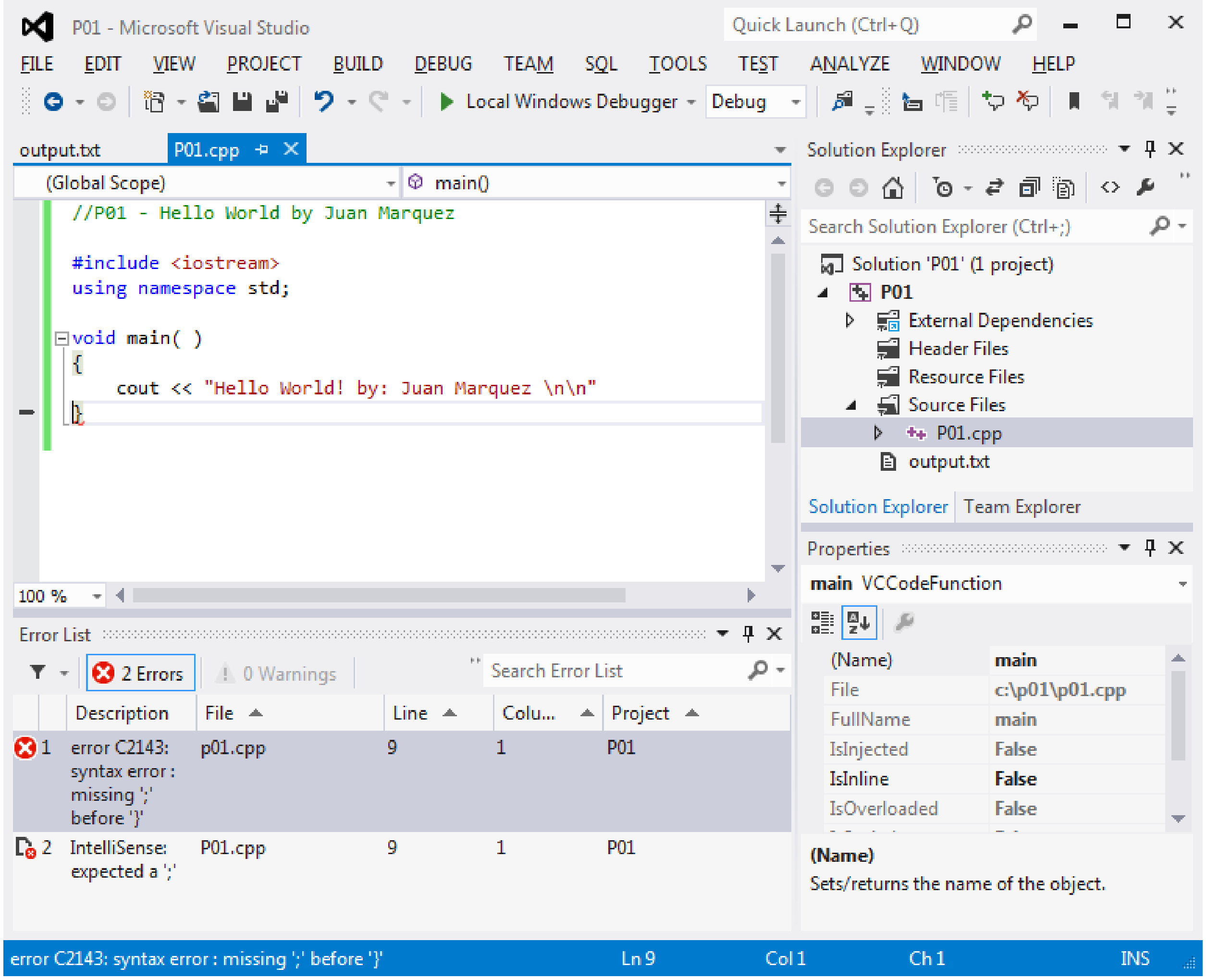
    error C2143: syntax error : missing ';' before '}'

  If IntelliSense is activated, it may also report a similar error.

    IntelliSense expected a ';'

g. Double click on the error on the Error List to highlight the error message and to move to the line where the error occurred.

  All statements must end with a semicolon, but the semicolon does NOT need to be on the same line. A semicolon is required somewhere before the closing brace, that is why the compiler reports the error on the line with the closing brace. However, to create neatly formatted programs we place the semicolon on the same line as the statement.



```
//P01 - Hello World by Juan Marquez

#include <iostream>
using namespace std;

void main( )
{
    cout << "Hello World! by: Juan Marquez \n\n"
}
```

	Description	File	Line	Colu...	Project
1	error C2143: syntax error : missing ';' before '}'	p01.cpp	9	1	P01
2	IntelliSense: expected a ';'	P01.cpp	9	1	P01

(Name)	main
File	c:\p01\p01.cpp
FullName	main
IsInjected	False
IsInline	False
IsOverloaded	False

(Name)

Sets/returns the name of the object.

error C2143: syntax error : missing ';' before '}' Ln 9 Col 1 Ch 1 INS

h. **Remove** the open quotation mark in front of Hello.

i. Select **Start Without Debugging** from the **Debug menu**.
Click on **Yes** if ask if you want to rebuild the project.
Click on **No** if asked if you want to run the last successful build.

Many errors should be reported on the Error List. You many need to scroll up and down in the Error List to see the messages.

j. **Double click** on the **first error** on the Error List to highlight the error message and to move to the line of code where the error occurred. The word Hello is not a C++ command, so the compiler reports it as error C2065: 'Hello' : undeclared identifier. Identifier is another word for variable, which will be covered in the next chapter.

k. Double click on the next few errors on the Error List to walk through a few errors. The line pointer does not move because all of the errors occurred on the same line. When there are many errors reported on the same line, and you noticed that you left something out that is very important, such as the opening quotation mark, it is best to add the quotation mark and recompile instead of walking through all of the errors.

l. Insert the quotation mark that was removed in front of Hello.

m. Insert the semicolon that was removed from the end of the cout command.

n. Select **Start Without Debugging** from the **Debug menu**.
Click on **Yes** if ask if you want to rebuild the project.
No errors should be reported.

14. To close project:

a. Click the **Save All** button.

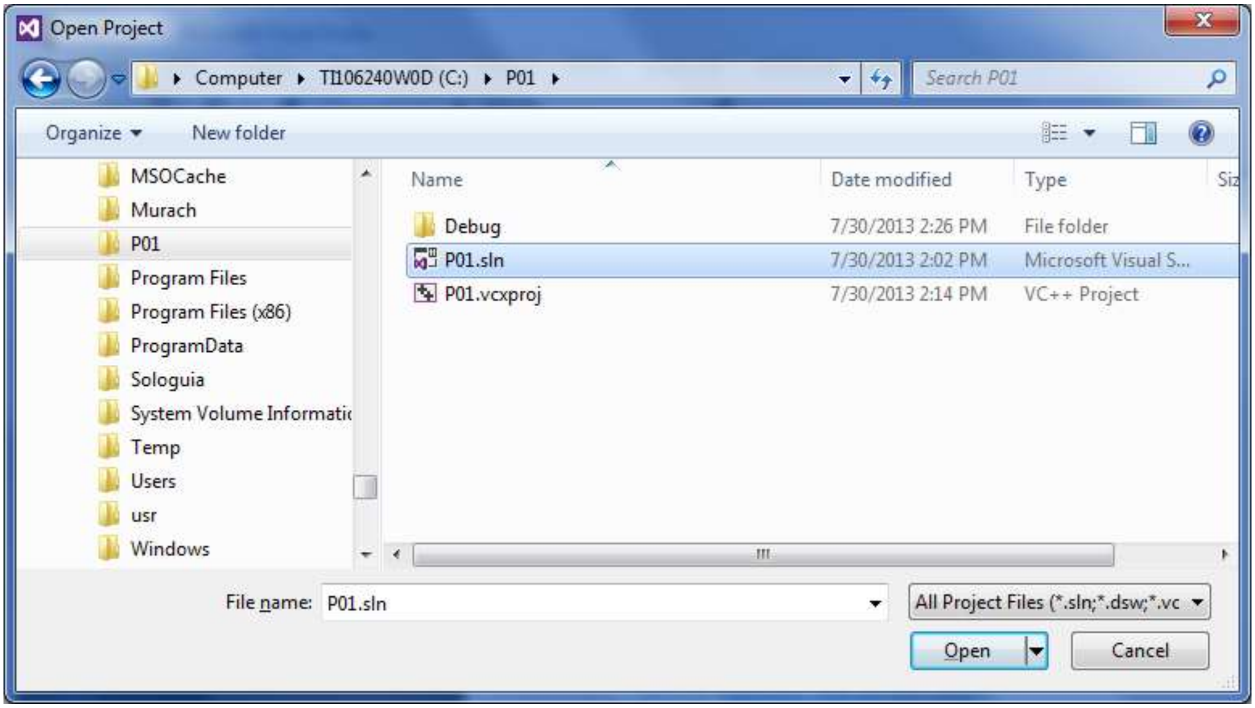
b. Click on **File** on the menu bar and select **Close Solution**.

15. To open an existing project:

    a. Click on **File** scroll down and mouse over **Open**, and then slide over and select **Project/Solution**.

    b. Navigate to the **P01 folder.**

    c. Click on the solution file (**P01.sln**) to select it, and then click on **Open**.



Name	Date modified	Type	Siz
Debug	7/30/2013 2:26 PM	File folder	
P01.sln	7/30/2013 2:02 PM	Microsoft Visual S...	
P01.vcxproj	7/30/2013 2:14 PM	VC++ Project	

File name: P01.sln

    d. You may also be able to double-click on the solution file when using Windows Explorer. **Always open the solution file (.sln) .** If you open the source code file (.cpp), Visual Studio will not know the projects settings. It will not be able to compile the program.

    Windows hides the files extensions by default. It may be best to search your help system to change the setting so that known file extensions are not hidden. It is usually set through Windows Explorer and selecting Folder Options from the Tools menu.

16. Click the **File** on the menu bar and select **Exit** to exit the program

17. **Submit online** the **source code** (P01.cpp) and **sample output** (output.txt) for P01 when completed. See Submitting assignments online.

18. If you are working in MCC's computer lab, it is very important that you **Eject** your storage device.