

Using Microsoft Visual Studio

CS 104 Teaching Team

Introduction This guide is intended to get you familiarized with C++ programming using an integrated development environment (IDE). Microsoft Visual Studio¹ has core project creation features including a code editor, debugger, and designer.

thing, but remember to keep it simple and easy to remember (e.g. Hello World)

1 Microsoft Visual Studio

1.1 Overview

Visual Studio is a program that allows you to write, edit, compile, and debug code in a broad range of programming languages. It is an Integrated Development Environment, which allows programmers to create a variety of projects, including user interface applications and websites. For the purposes of this class and this guide, we will focus on the basic usage of Visual Studio to create console programs using C++.

1.2 Creating a Project in Visual Studio

For this exercise, we will be printing out the phrase, Hello World! To begin writing a program in Visual Studio, you must first create a new project.

1. At the top left corner, click **File**. Select **New**, and then select **Project**. A dialog box will appear.
2. In the left pane of the dialog box, click on **Win32** under the Visual C++ section.
3. Select the Win32 Console Application option, and name your project. You can name it any-

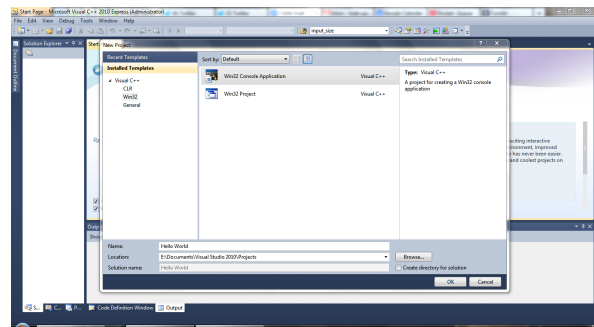


Figure 1: Project Creation Dialog Box

4. On the Application Settings page under Additional Options, check **Empty Project**. There should be a message at the bottom that says Creating project Hello World project creation successful. This will ensure that your program does not have any initial miscellaneous code that could possibly affect the code compilation.

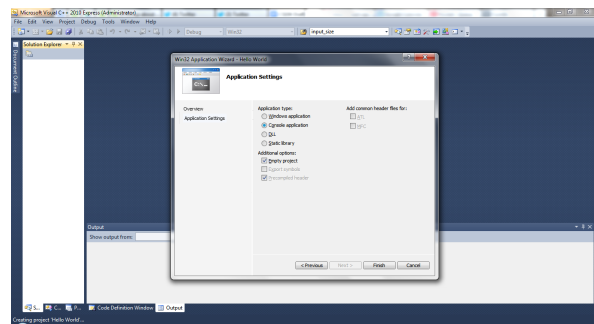


Figure 2: Application Settings

¹<http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-windows-desktop>

1.3 Writing Code for your Project

1. Your project will appear on the screen. To begin writing code for your program, you must add a source file. To do so, right click **Source Files** under your project name. Select **Add New Item**.

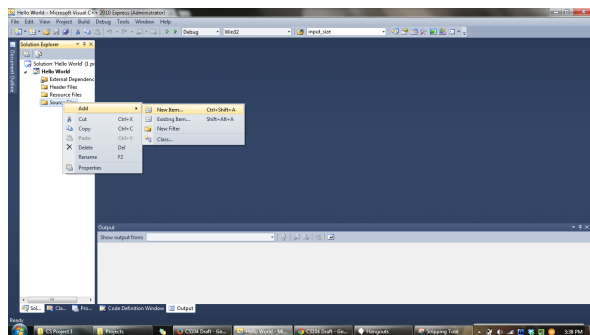


Figure 3: Adding a Source File

2. A dialog box will appear. Under Visual C++ on the left side, click **Code**. Select **C++ File** and name your source file. Most people use **main** to indicate where the core of the program code sits.
3. A blank page will appear. For your main source file, you must include the following lines of code:

(a) `include <iostream>`

- i. This command includes the library where all input and output functions are defined. That way, when the program is being built, the compiler will know what each input and output function means.

(b) `using namespace std;`

- i. As you will learn in class, a namespace groups items such as objects and classes under a single name.² The `std` namespace is used by all files in the C++ library, so it is necessary to include this line of command to avoid

²<http://www.cplusplus.com/doc/tutorial/namespaces/>

having to write `std::` before each command.

(c) `int main() {..your code here..
return 0;}`

- i. This is where your core program code will sit. The code for the program will be between the curly brackets. At the end of your code, you must return an integer value (usually 0) to instruct the program to stop when it has completed running through the code.

4. To print out a string, use the command `cout`, followed by `<<`. Put the desired string, which is in this case `Hello World!`, in double quotation marks.

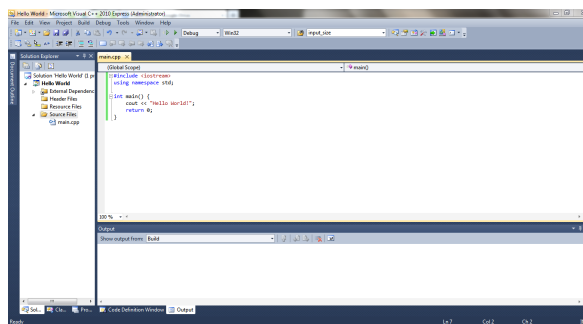


Figure 4: Main Source File

1.4 Building Your Project

To see how your program works, you must build the code. This includes compilation and execution of the project.

1. Click on **Build** at the top of the screen. Select **Build Solution**.
2. If your project compiled successfully, the dialog box at the bottom of the screen will say **Build: 1 succeeded, 0 failed** and a terminal will pop up.
3. If your project did not compile successfully, make sure that all of your syntax is correct (e.g. no missing commas or semicolons). You can also

debug your code to see where your program fails to build.

2 Simple Debugging in Visual Studio

Visual Studio has a built in debugger. A debugger allows you to set break points in your code that will allow you to see where your code may not function the way you intended. When you begin the debugging process for a specific program, the code will execute up until the statement where the break point has been set. The break point statement will not be executed.

1. To set a break point, click on the tray on the side. A red bubble should appear where you clicked your mouse. If you right click on the tray, you will see options for different break points. There are several types of break points. One type is a conditional break point, which will stop the program when that condition is true.

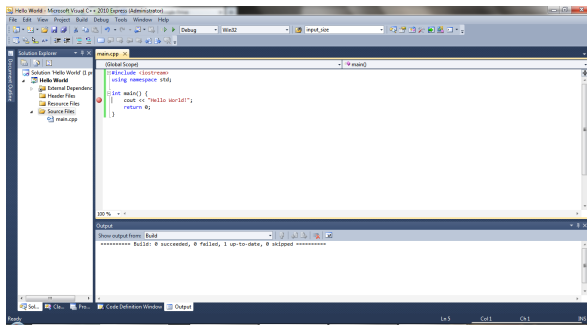


Figure 5: Debugging in Visual Studio

2. Right click on your project name, and then select Debug.