

Downloading and Installing DevC++

What is DevC++?

DevC++ is a full-featured integrated development environment (IDE) that uses the MinGW version of C++. An IDE is a program that allows you to edit, compile, and debug your C++ programs without using the command-line.

DevC++ comes with everything you need to compile and link both console-mode and GUI programs that will run on Windows 95/98/NT/2K/XP; your programs will **not** run on Windows 3.1 or on plain DOS machines however. You don't need to download and install anything else.

The rest of this page will walk you through the steps necessary to download, install, and run your C++ programs using DevC++ and the included version of the MinGW C++ compiler.

What is MinGW?

MinGW ("Minimalistic GNU for Windows") is a command-line C/C++ compiler and utilities, based on the GNU GCC project. It compiles and links code to be run on Win32 platforms (Win95 through XP). Unlike some other versions of GCC, MinGW uses the Microsoft runtime libraries, distributed with the Windows operating system. Since these runtime libraries are not distributed using GNU's General Public License (GPL), you do not have to distribute your source code with your programs unless, of course, you use a GPL library in your programs.

Step One: Download DevC++

Click on the following link:

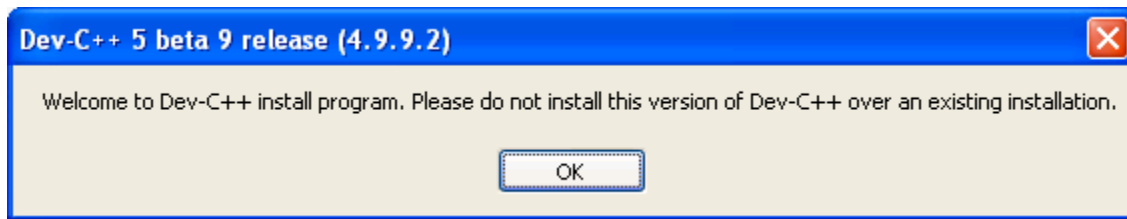
<http://www.bloodshed.net/dev/devcpp.html>

At the bottom of the page, select to download: ***Dev-C++ 5.0 beta 9.2 (4.9.9.2) (9.0 MB) with Mingw/GCC 3.4.2***. This will take you to a download page. Save the file, ***devcpp-4.9.9.2_setup***, to your computer.

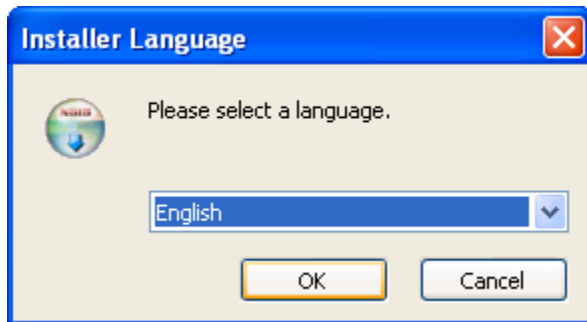
The downloaded file is about 9 MB; on a DSL/Cable connection it takes under 5 minutes to download. On a dial-up connection, though, downloading may take a while. In that case, you may want to copy the installation files from the computer lab, then transfer it to a USB thumb-drive.

Step Two: Install DevC++

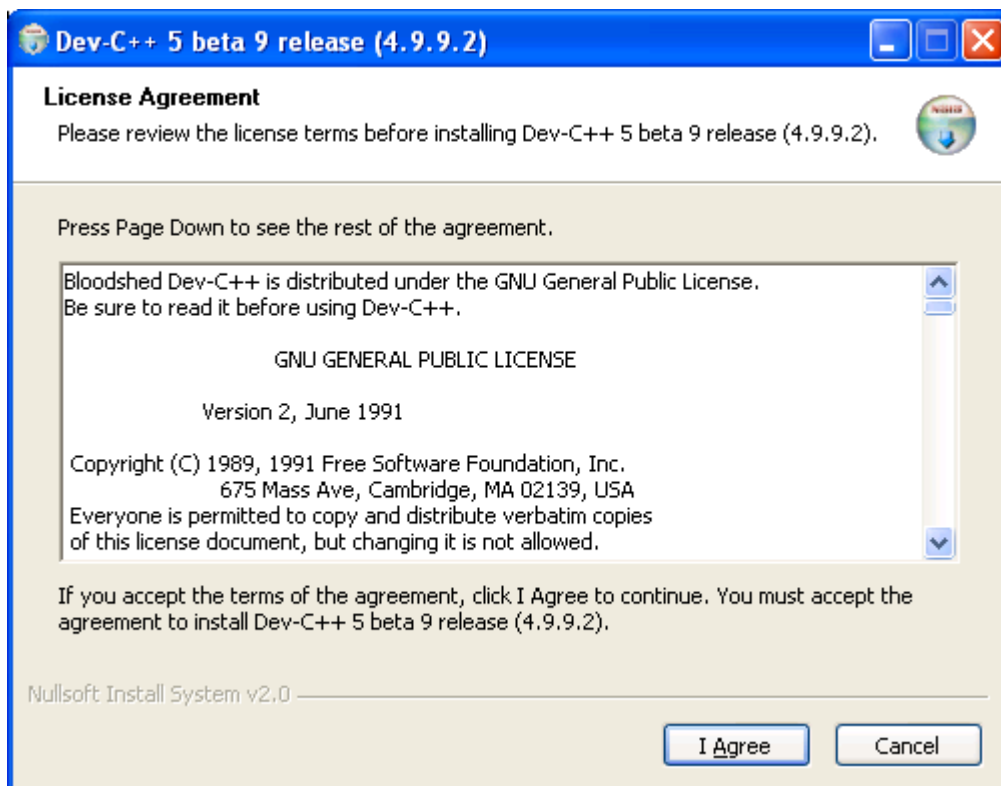
Locate the file ***devcpp-4.9.9.2_setup*** on your computer.



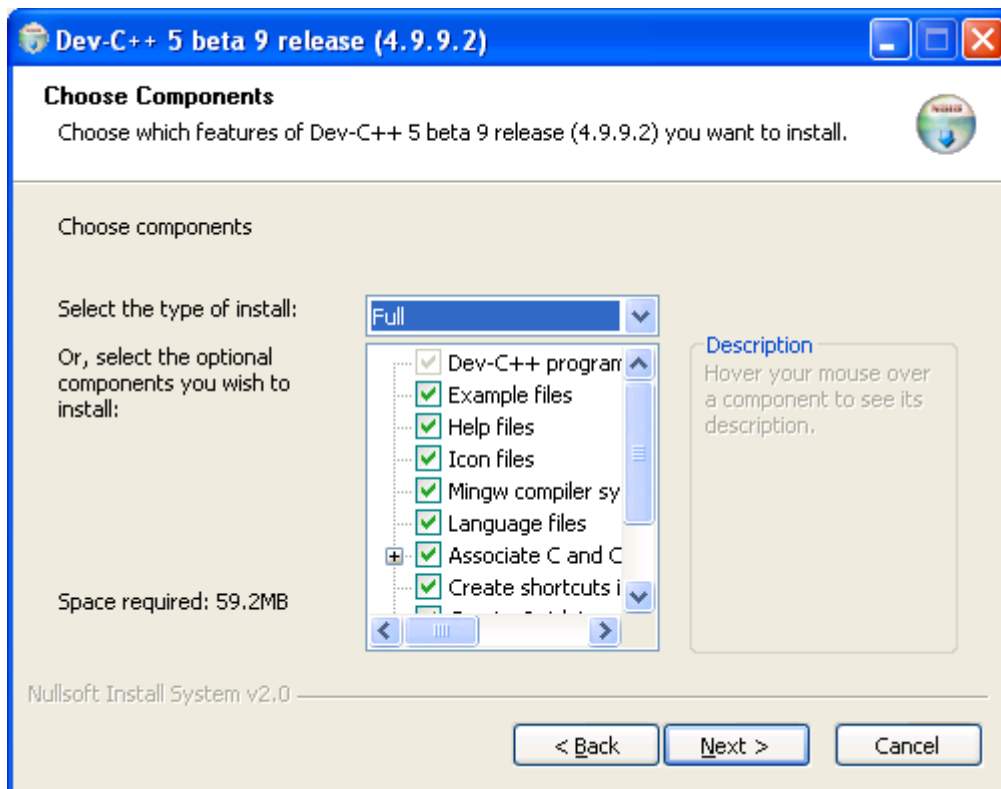
You will get a window indicate the start of the install followed by:



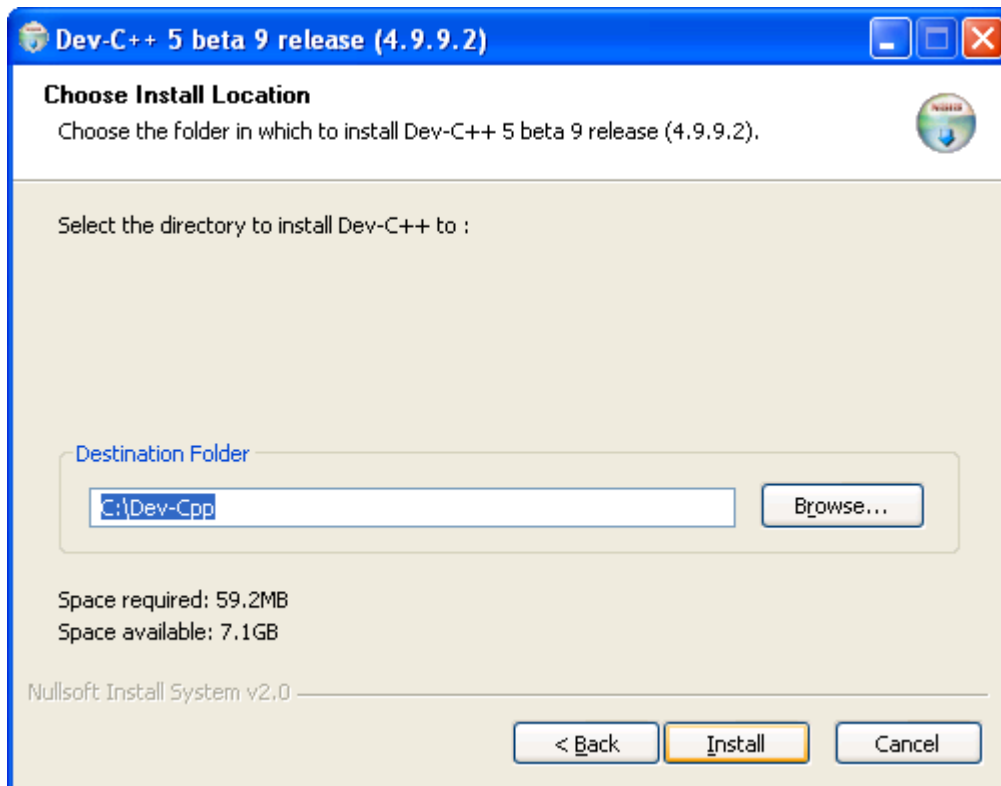
Select "OK" for English.



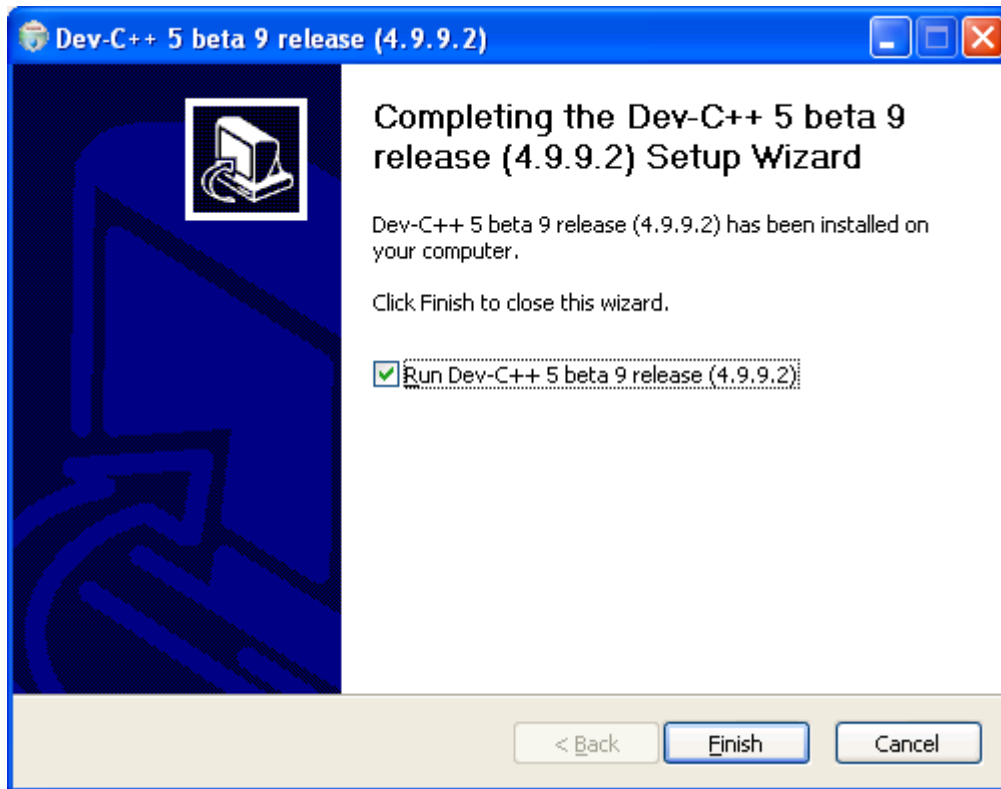
Select "I Agree"



Select "Next".



Select "Install".

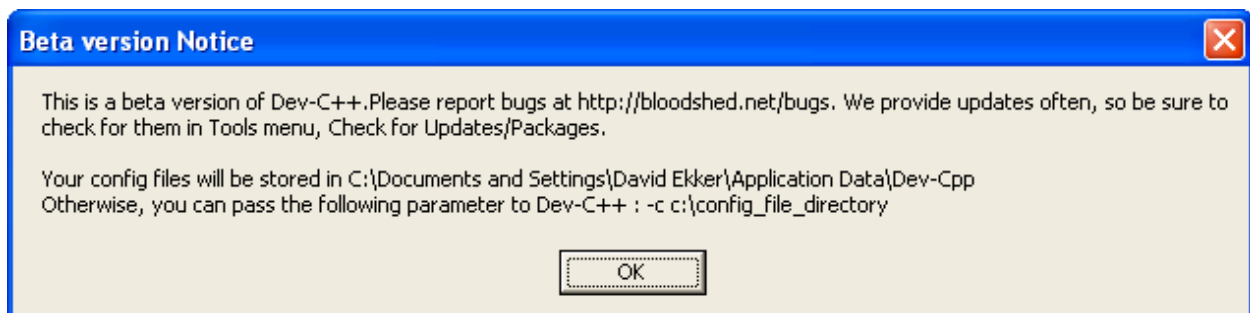


DevC++ 5 beta 9 release (4.9.9.2) should be installed.

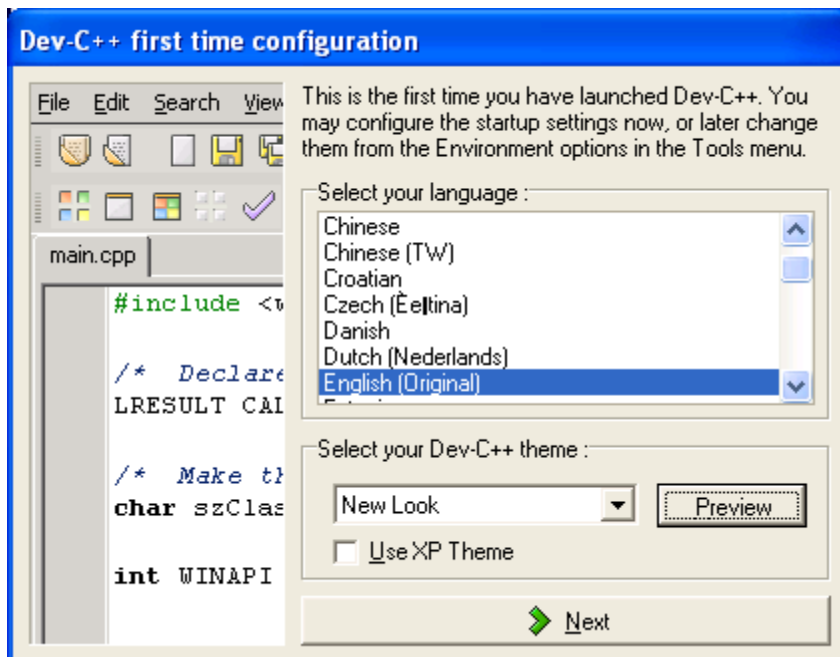
Step Three: Configuring the IDE

A. Configuring the IDE for the first time

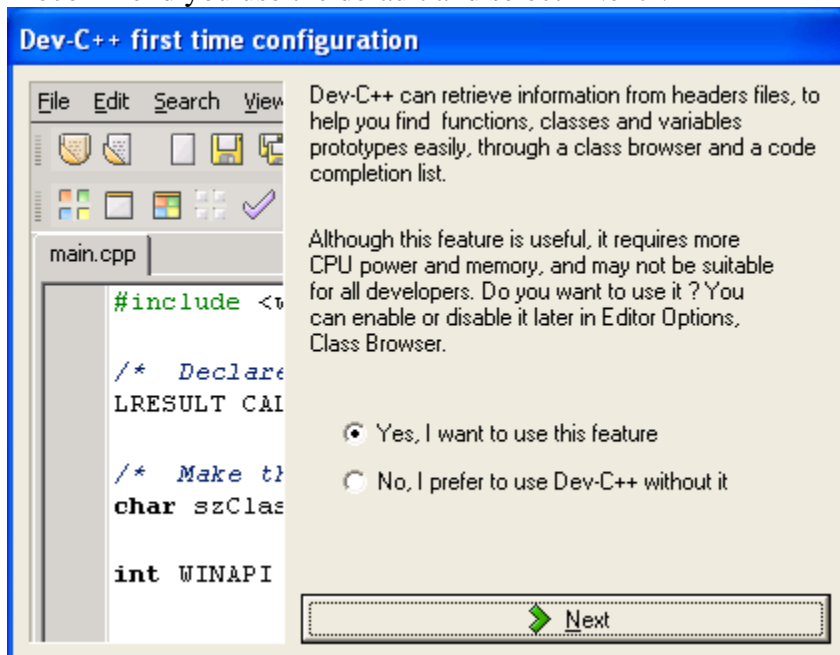
You will need to configure the IDE the first time you run DevC++. You may also need to configure it the first time you run it in the computer lab also.



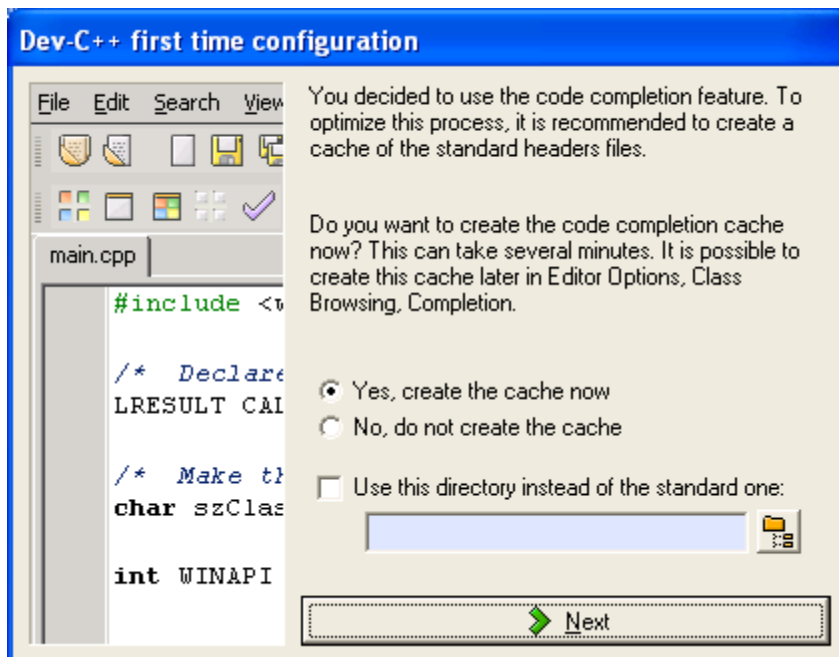
Select "OK".



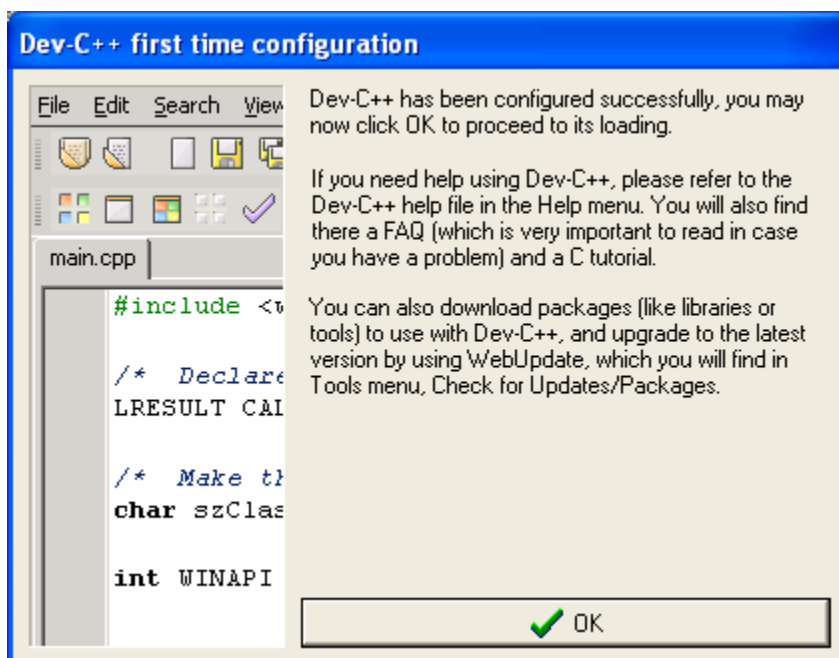
I recommend you use the default and select “Next”.



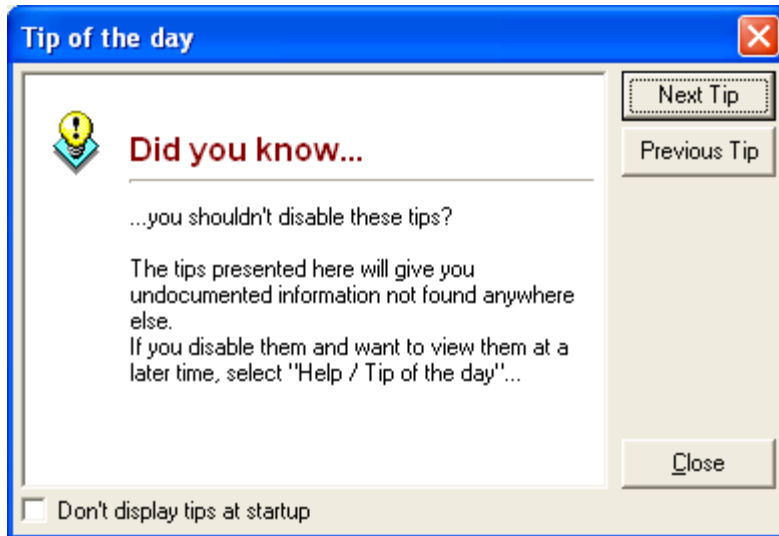
For most users this is not a problem, select “Next”.



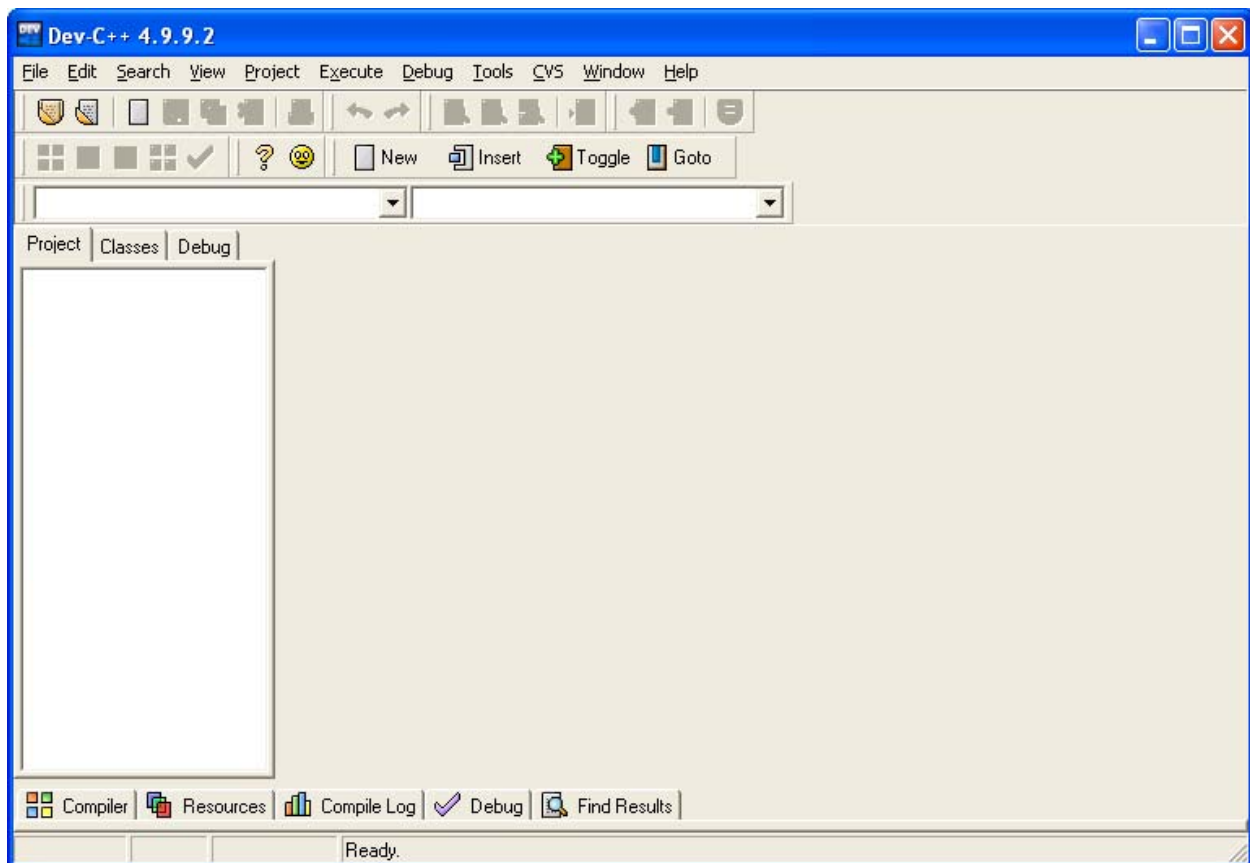
Select “Next”. After about 2 minutes, you should get:



Select “OK”. Dev-C++ should start and you will get a “Tip of the day” window.

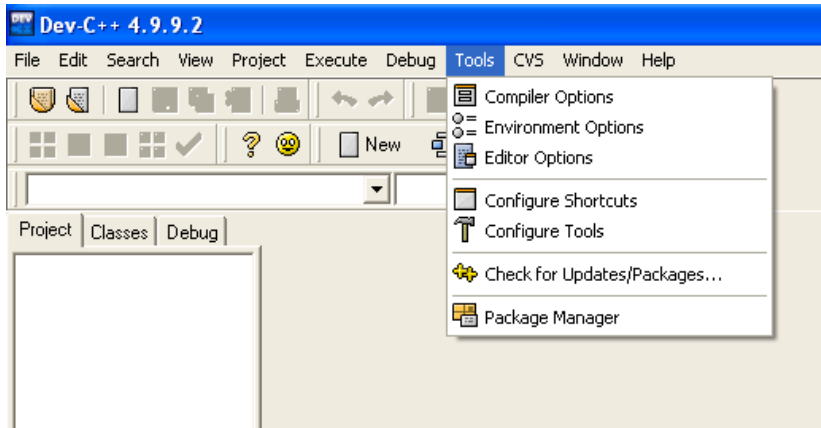


Read them if you like, when done, select “Close” and you should get the following window:

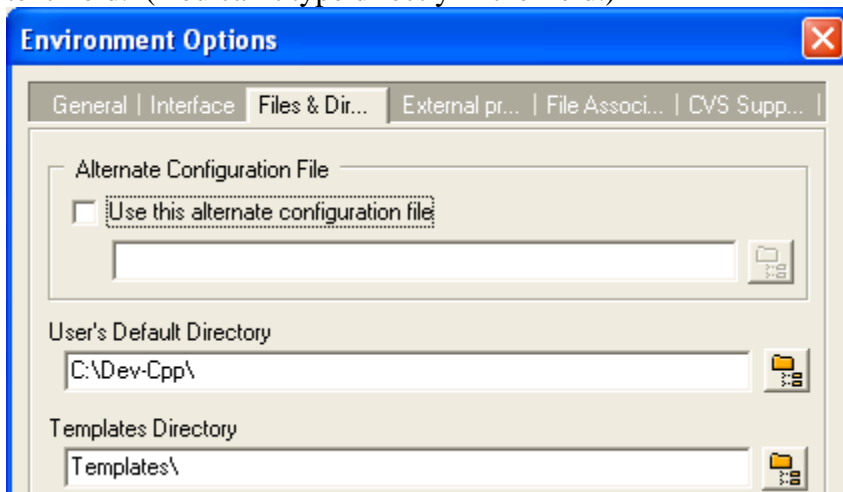


B. Configure the Environment Options

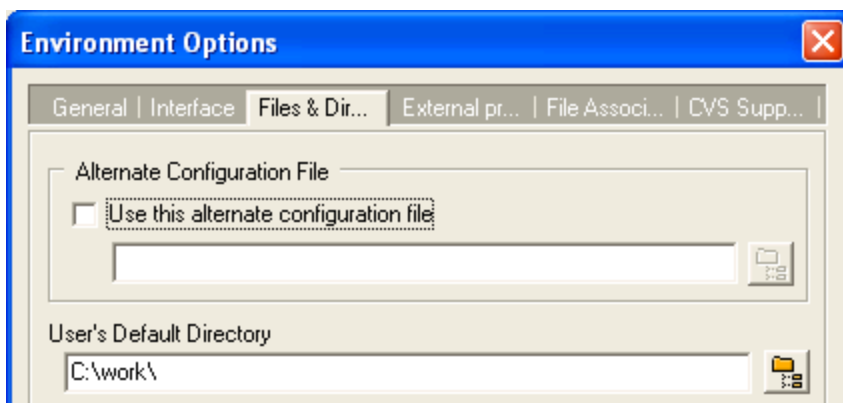
Environment options make it easier for using the compiler. The Environment Options are located under the Tools menu:



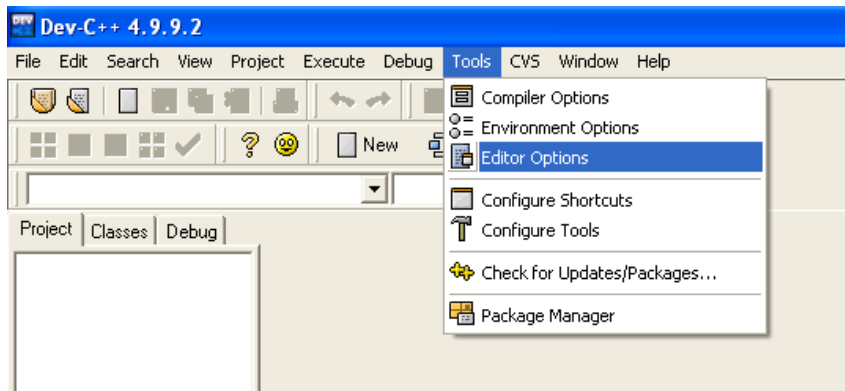
Change to the Files & Dir tab, and then click the button to the right of User's Default Directory text field. (You can't type directly in the field.)



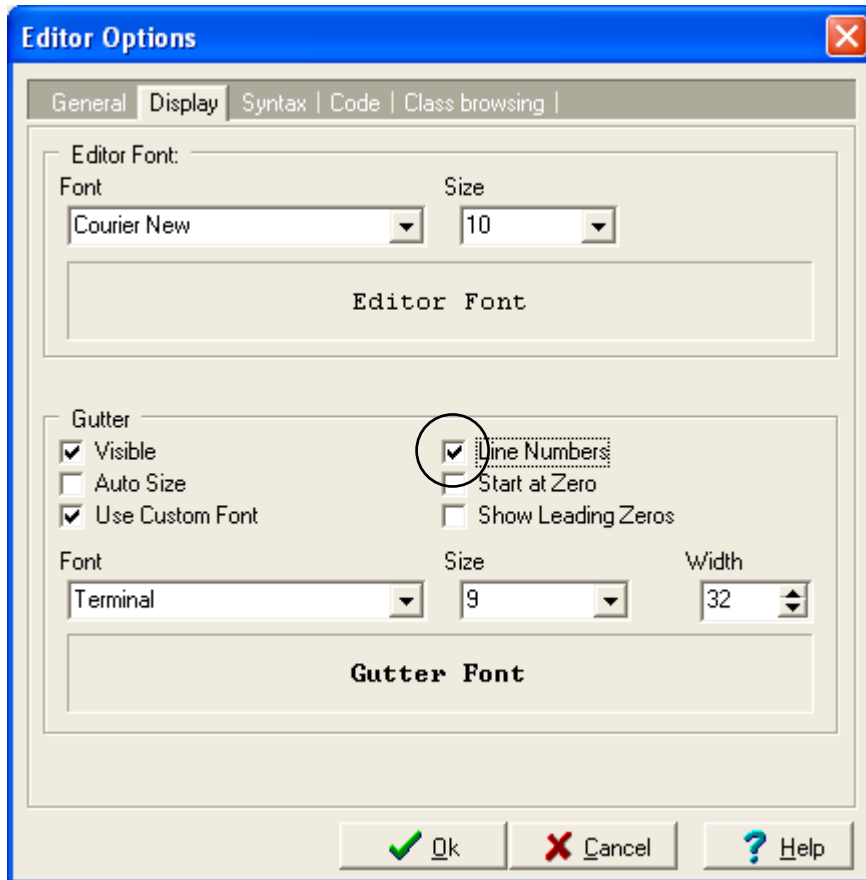
Locate the folder where you want to store your files. I like storing them on my c: drive in a folder called work. It makes it easier when working on the individual source files we will be using at the beginning of the class:



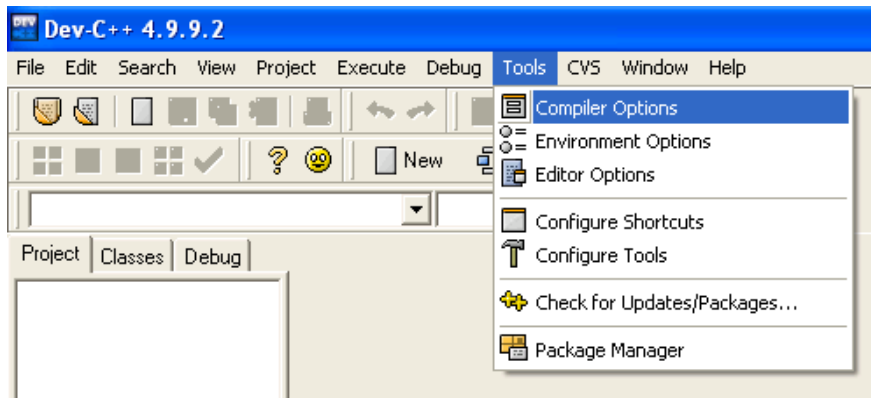
C. Configure the Editor Options



Feel free to experiment and select whichever options you like. The one I want is to have line numbers displayed. To do this, select the **Display** tab, and click the **Line Numbers** checkbox, as shown here:



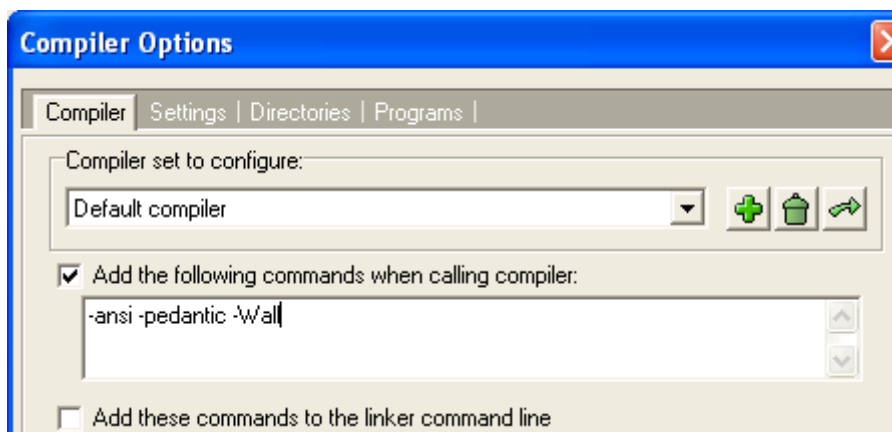
D. Configure the Compiler Options



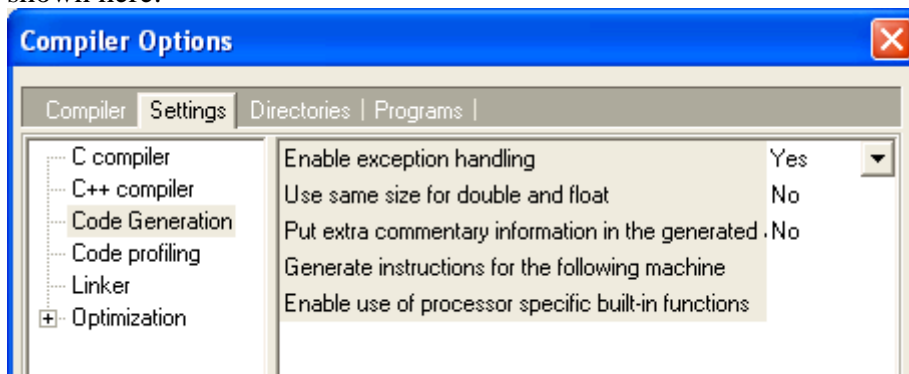
On the **Compiler** tab, add the following switches when calling the compiler:

```
-ansi -pedantic -Wall
```

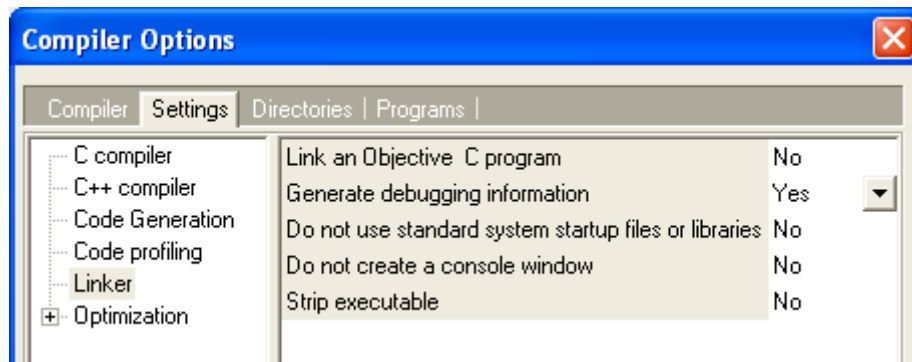
Note that the W in Wall is capitalized, but everything else is lowercase. Here's what this tab should look like:



On the **Settings** tab, choose **Code Generation**, and change *Enable exception handling* to **Yes**, as shown here:



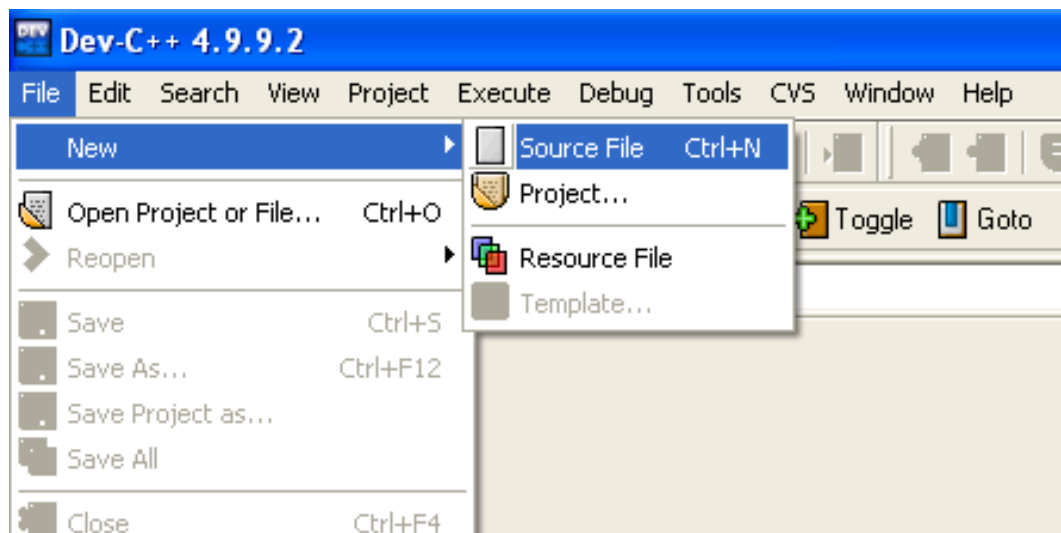
While still on the **Settings** tab, choose **Linker** and change *Generate debugging information* to **Yes** as shown here:



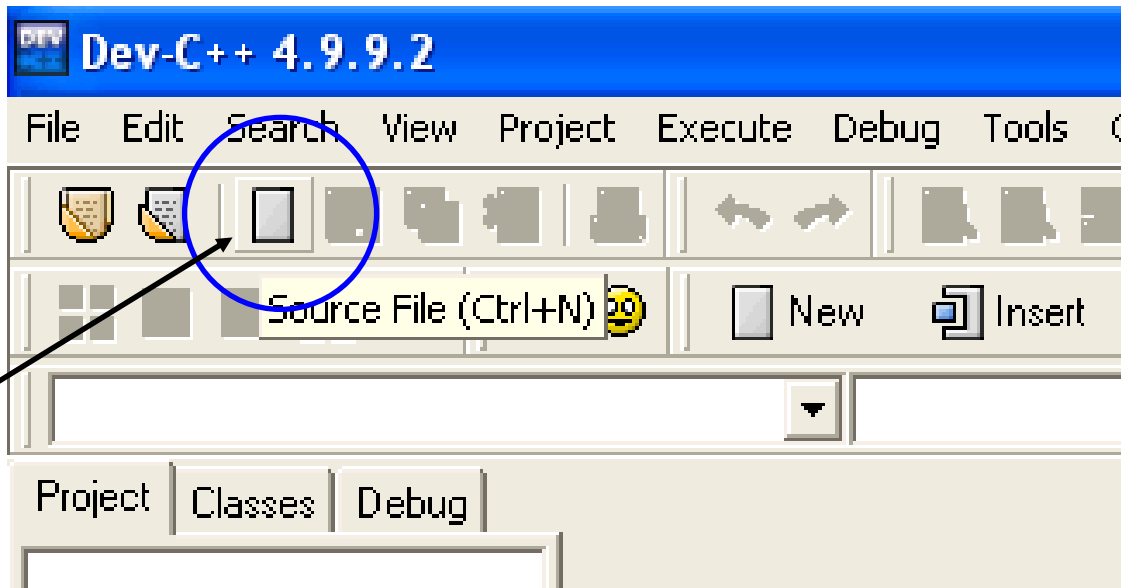
Step 4: Create source file, compile and run your first program.

A. Open a new Source File

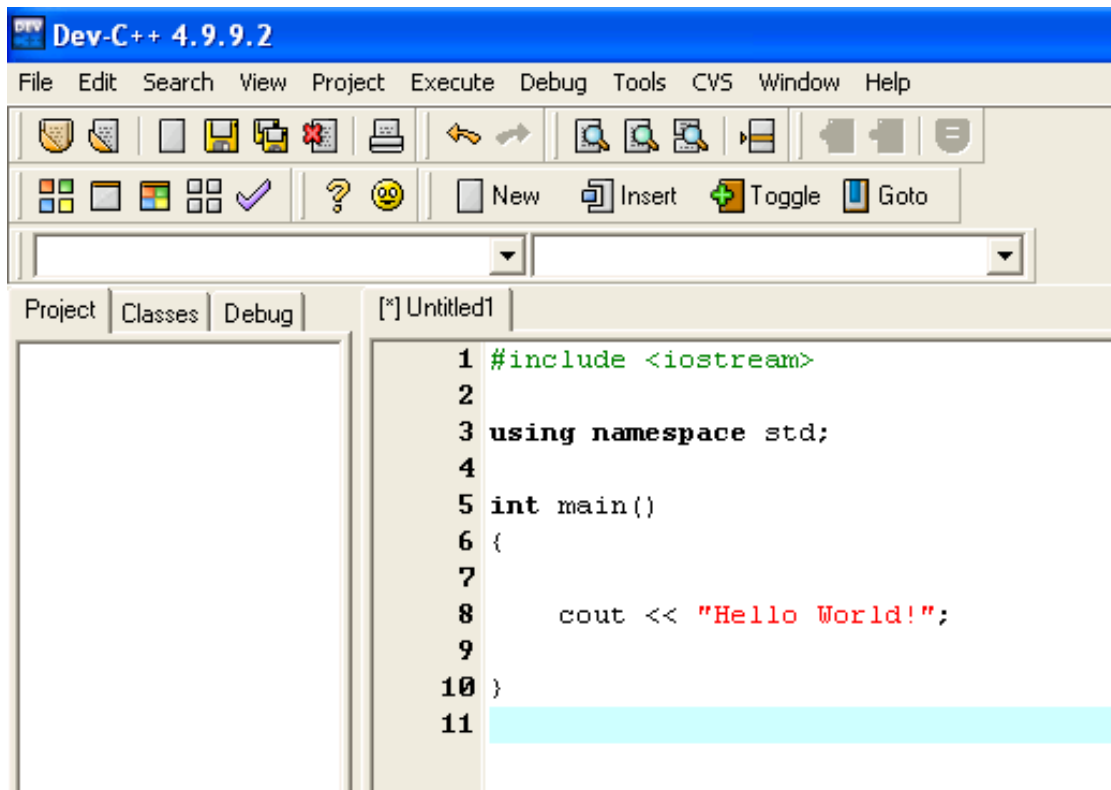
Using the File Menu,



Or simply by clicking on new Source File button or pressing **Ctrl+N**:



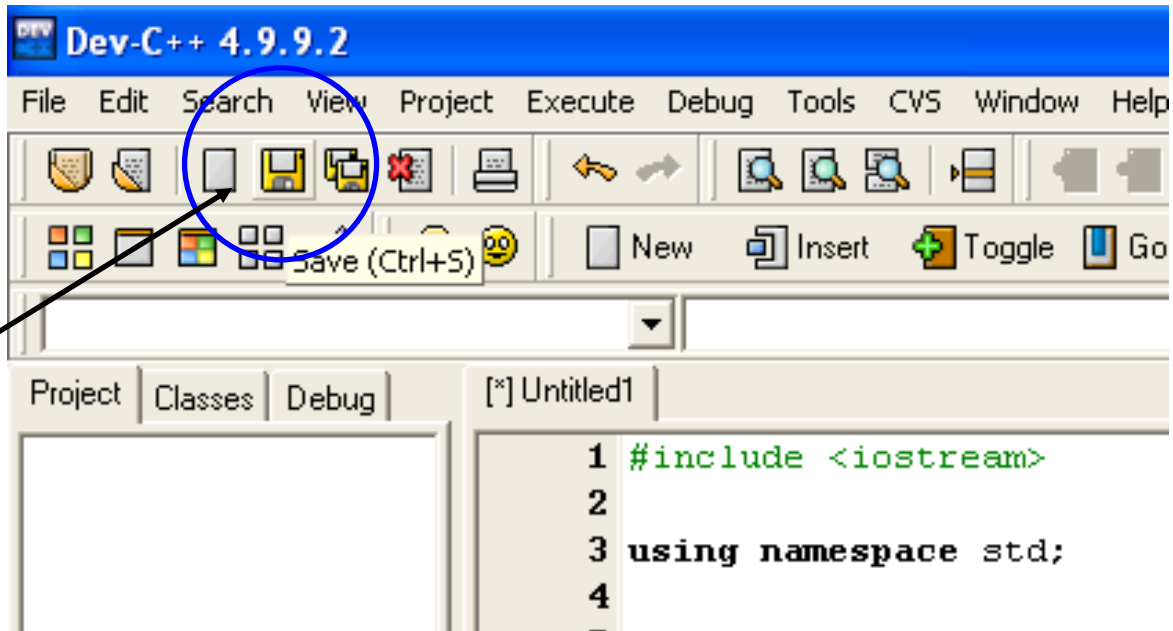
Carefully type in the following program into the resulting window, be careful, it is case sensitive!



Make sure you have everything exactly as given.

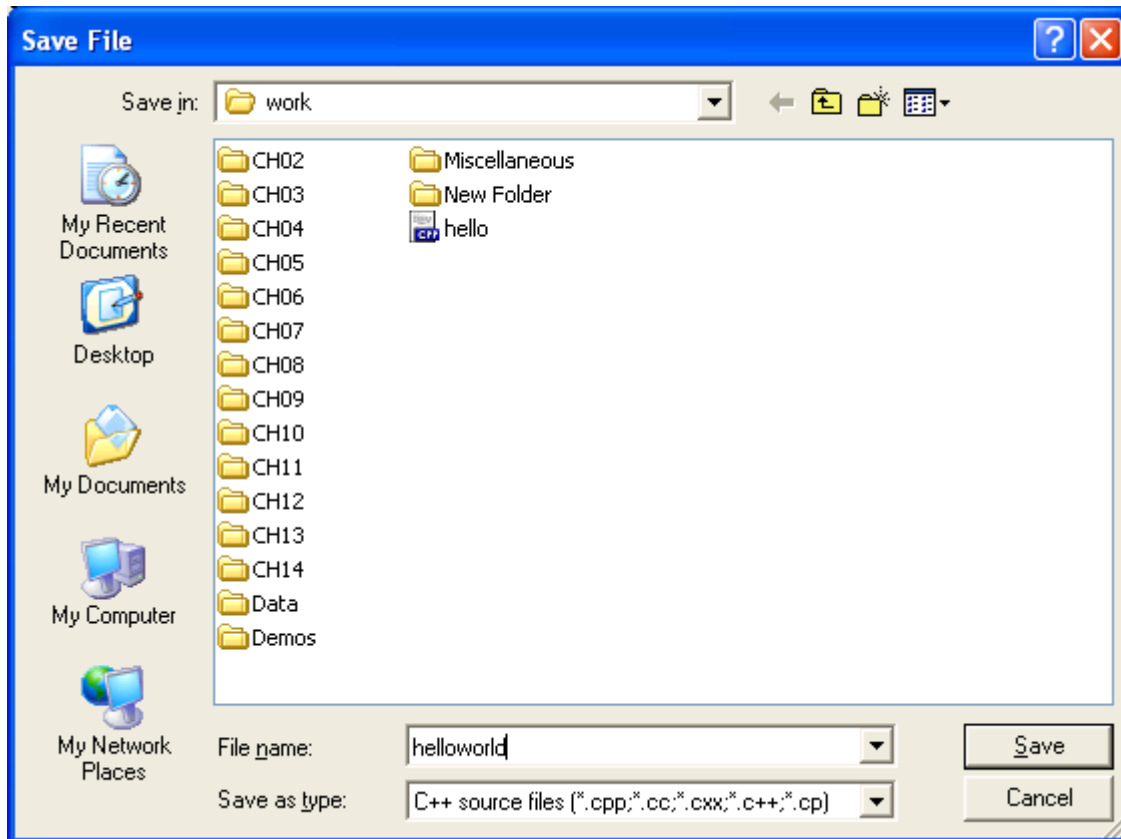
B. Saving your source file

Once done, go ahead and save the file:



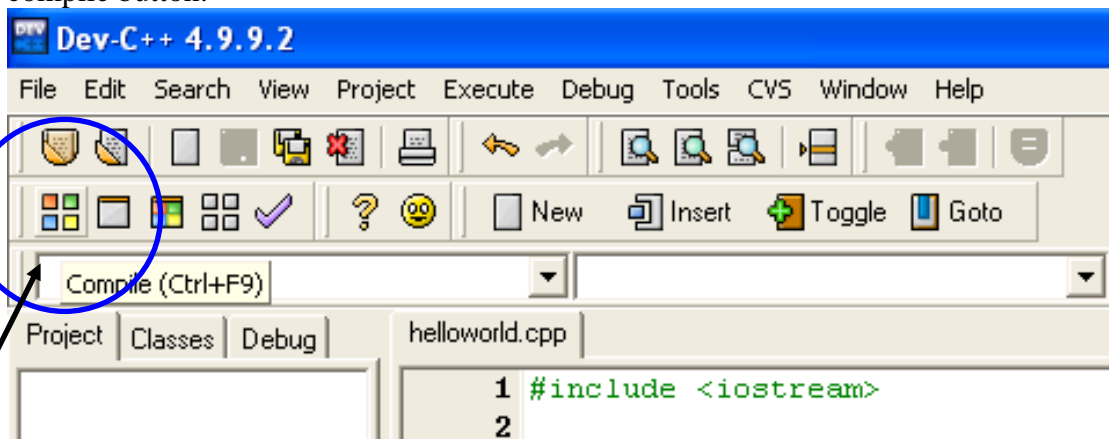
When saving files, make sure you follow the following rules:

1. File names should start with a letter, it is not case sensitive!
2. File names may include letters, numbers, or under_scores, but the first character must be a letter!
3. File names should be descriptive. At the beginning of the class, there is not much problem, but as we get close to the end of class, it will be more important.

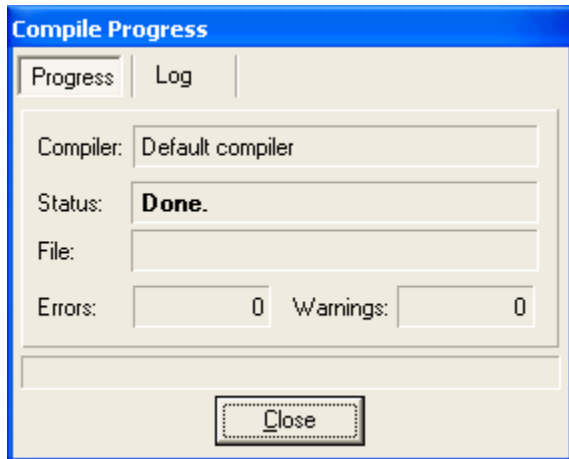


C. Compiling your source file

For simple programs, only one file is required and you can compile your source file using the compile button:



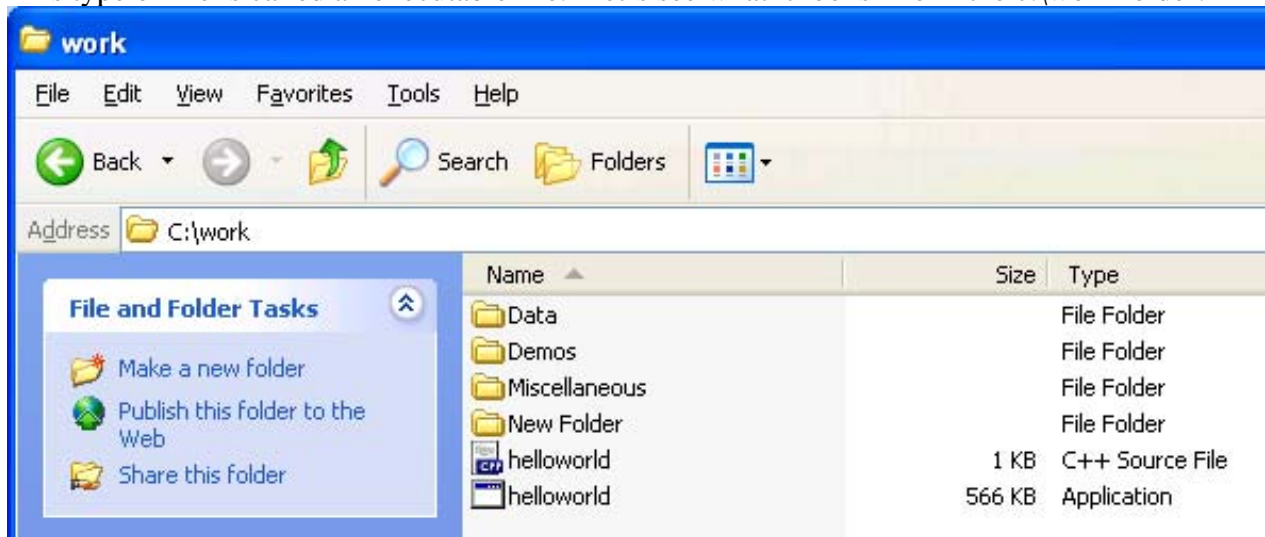
If you compiled successfully, you should get the following window:



You have successfully compiled your first program!

D. Running your program

The compiler took your source code and converted it into a file which will run on your computer. This type of file is called an executable file. Let's see what it looks like in the c:\work folder:



There are two files named “helloworld”. The first is the file you created by typing in code and saving. The second is the executable file created by the computer. This is an actual computer program. If you double click on it, a window quickly appears and disappears. This is MS Windows “Command Prompt” window. The program opens a Command Prompt window, runs your program, and then **immediately** closes the window before you can examine the output.

You have two ways of addressing this situation:

1. Adding a system(“pause”) command – recommended option
2. Using the Command Prompt window – not covered here

1. Adding a system (“pause”) command

Carefully modify the source file to add a pause in the program:

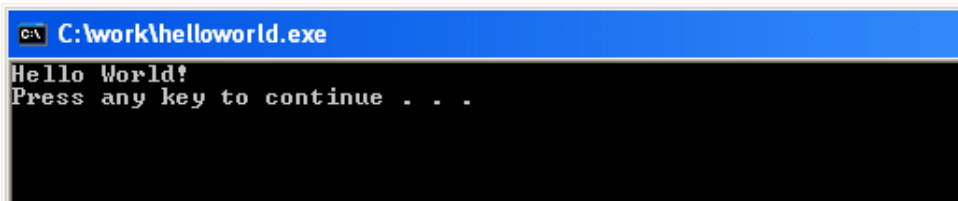
```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World!" << endl;

    system("pause");
}
```

Now compile and run the program. You should get the following window:

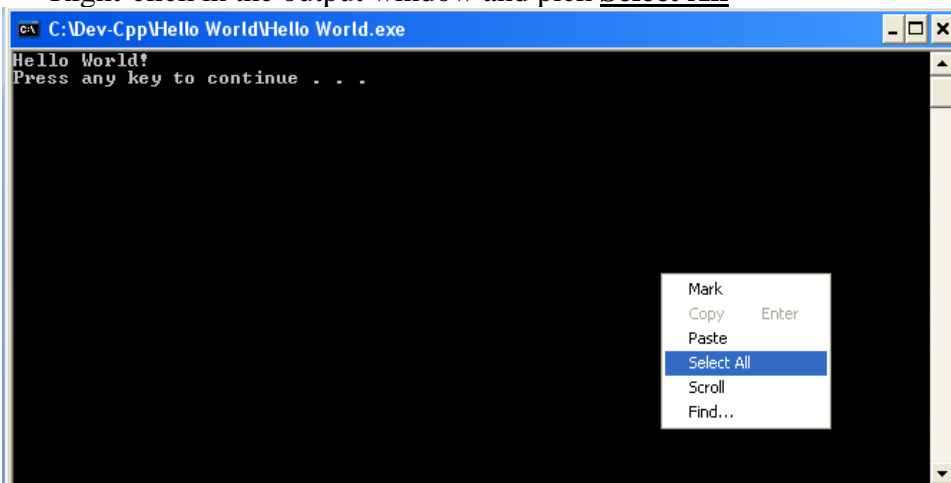


```
C:\work\helloworld.exe
Hello World!
Press any key to continue . . .
```

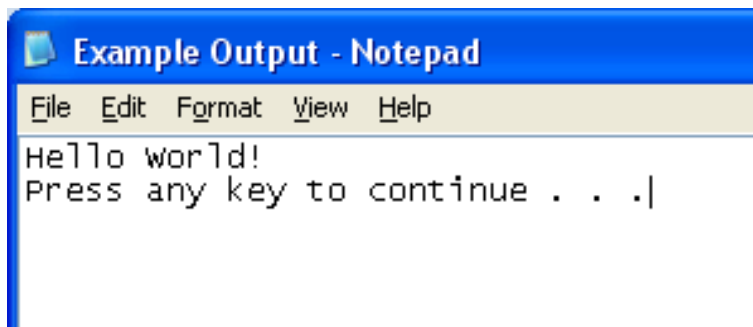
Press any key to continue.

Printing from the output window

- Right-click in the output window and pick **Select All**



- Press the **Enter** key to copy the selection to the clipboard.
- Open Notepad (or any word processor) using **Start – All Programs – Accessories – Notepad**
- Paste the clipboard contents into clipboard
- Select **File – Print** within Notepad.
- If you wish to save the output file, use the File – Save option in Notepad.



Fonts: Recommended font: Courier New

- When you paste the contents of the output window into Notepad, some of your formatting may seem to change (this is especially noticeable with an output table). This is because many fonts use different widths for different letters.
- **Courier New** is a font that uses the same amount of spacing for each letter or symbol which is nice when columns of numbers are created as in tables.
- After you paste the contents of the output window into Notepad, change the font to **Courier New** if you don't like the appearance of the output.