* **Dev C++**

Dev-C++ is an IDE (Integrated Development Environment) for C++ programmers. An IDE is a software application that contains a source code editor, a compiler, and a debugger, all in one. This documentation shows you how to download an easy-to-use IDE called DEV-C++.

* **Requirements:**

- Windows 95 or higher.

- 32 MB of RAM.

- The executables compiled by Dev-C++ will need MSVCRT.DLL (comes with Windows 95 OSR 2 or higher).

* **Features of Dev C++**

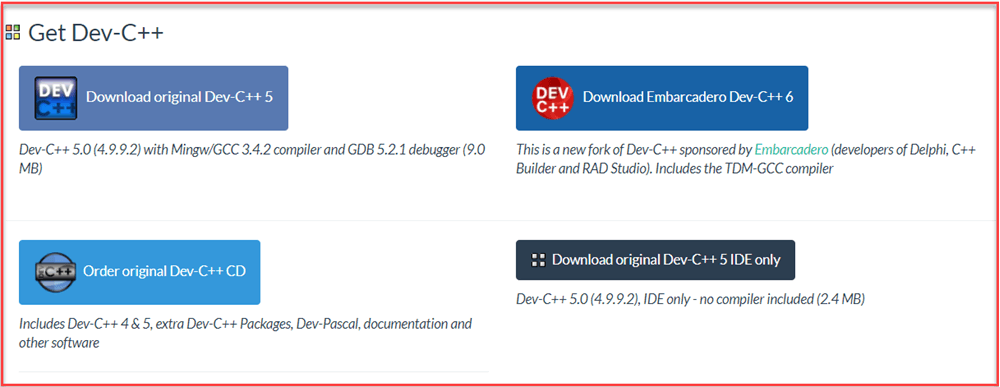
Here are some important features of Dev-C++ IDE:

* Dev-C++ IDE allows us to use integrated debugging using GDB.
* Localization feature that provides support for multiple languages.
* Offers editing and compiling the resource files.
* It has a inbuilt find and replace facility.
* Helps you to create your own project templates to create project types.
* It offers support for class browser as well as debug variable browser.
* Provides project manager feature that helps you to manage various projects.
* Provides CVS support for source code management.
* **Installation**

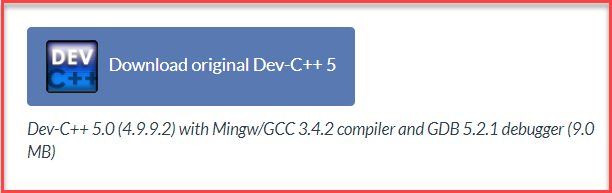
**To install Dev C++ software, you need to follow the following steps.**

**Step 1)** First you must download the Dev C++ on your Windows machine. Visit to Download Dev C++: <http://www.bloodshed.net/>

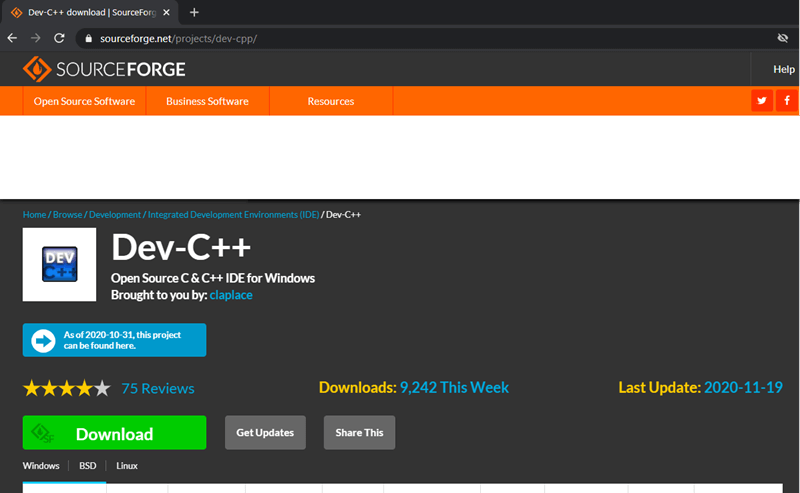
**Step 2)** There are packages for different Operating Systems.



**Step 3)** Under package Dev-C++ 5.0 (4.9.9.2) with Mingw/GCC 3.4.2 compiler and GDB 5.2.1 debugger (9.0 MB) Click on the link “Download from SourceForge”.



**Step 4)** This package will download C++ **.exe file** for Windows that can be used to install on Windows 7/8/XP/Vista/10.

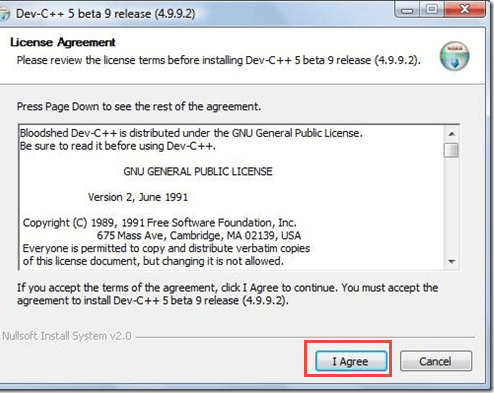


**Step 5)** You will direct to Source Forge website, and your C++ download will start automatically.

* Click on save button to save. By default, it is saved in “Downloads” folder.
* After the download completes, go to the saved .exe file and click on it to Run.
* The installer will ask you a language to select. Select “English” and click on “OK”.

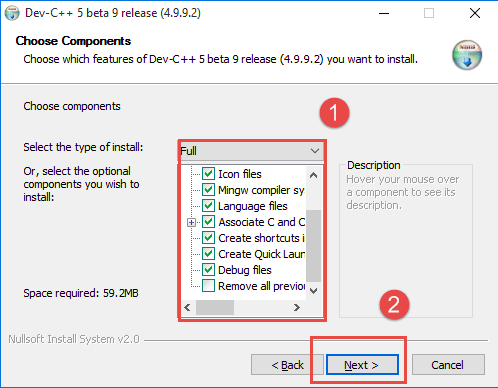


* Then screen for license agreement will appear. Click on “I agree” to proceed further.



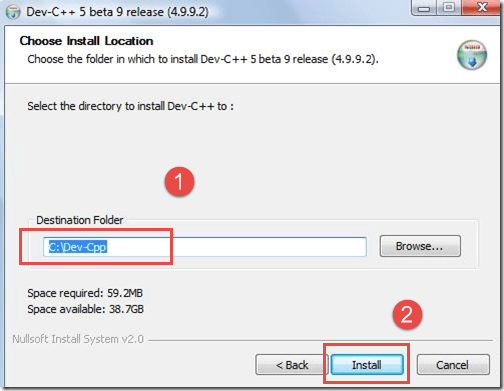
**Step 6)** In this step,

1. You can see different components of Dev C++ that will be installed with this package.
2. Just click on “next” button.

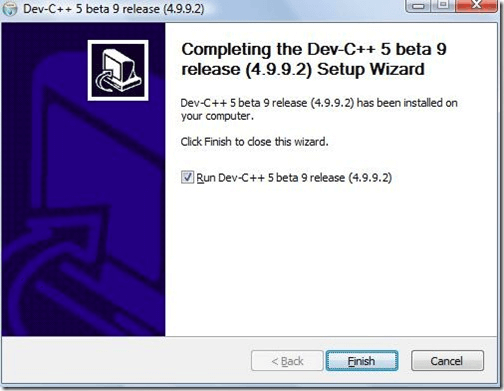


**Step 7)** In this step,

1. By default, the destination folder is in C drive. You are free to change this destination folder but make sure you have enough memory.
2. Click on “Install” button.



Now, Dev C++ is installed successfully on your Windows. Select Run Dev C++” to run it and click on ” Finish” button.

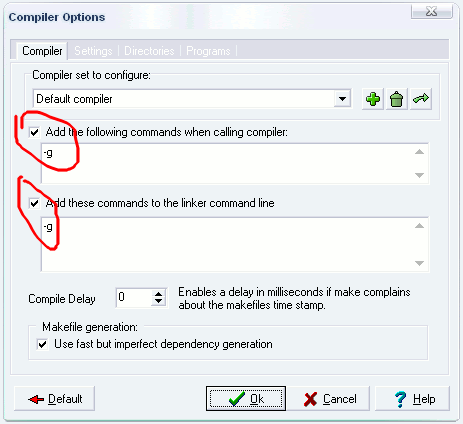


* **Debugging in Dev C++**

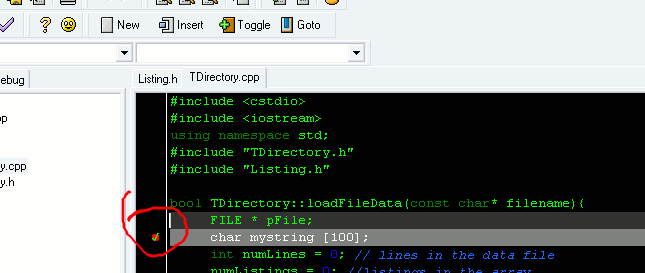
You can use the Dev-c++ debugger to help you find errors in your program. In debug mode, you can step through the execution of your program one line at a time. Also, you can stop at any time to see the value of your variables. Here's how:

1) Normally, you can debug your program just by clicking the icon that looks like this:

2) Sometimes, you see a message that says "Your project does not have debugging information, do you want to enable debugging and rebuild your project?" If this message continues to appear even after you click "yes" several times then you can do the following: Click Tools->Compiler Options. Then add '-g' as shown below:



3) To make your program stop at a certain line in your code, add a breakpoint at that line by clicking in the space on the left of your code**:**

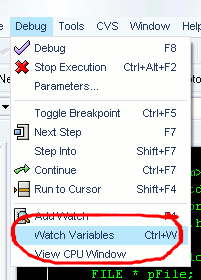


4) Now when you click the debug icon your program will stop at the line you marked as a breakpoint.

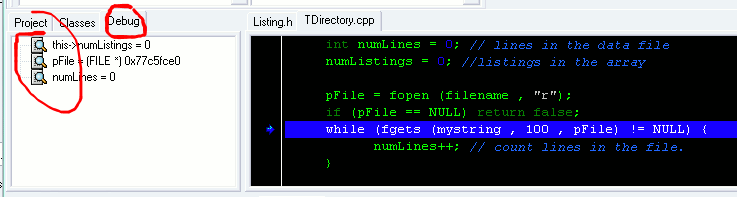
5) To step through your code one line at a time, click your cursor on a line in your code where you think you have an error, click "run to cursor", then use this:



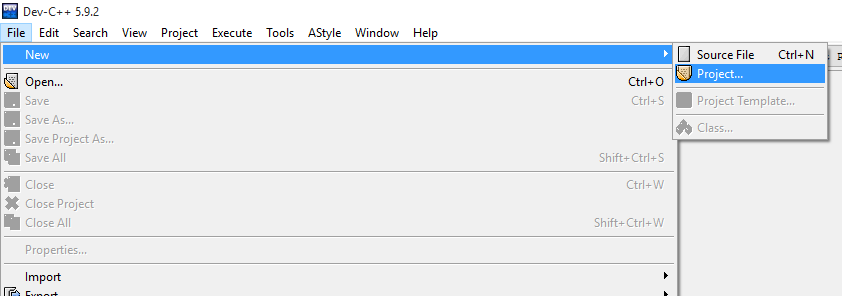
6) At any time, you can see the value of your variables using the "watch variables" function:



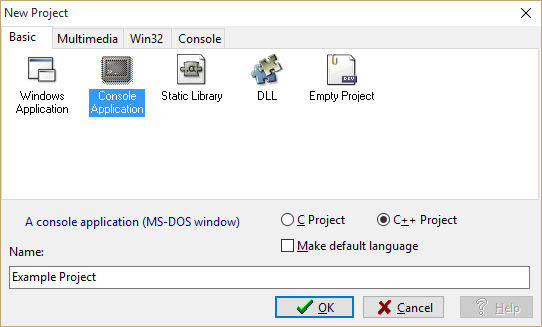
7) After you have selected "Watch Variables" you can hover your mouse over any variable name in your code. Rest your mouse over the variable name for a few seconds and then you will see the variable and its value appear on the left side under the 'debug' tab:



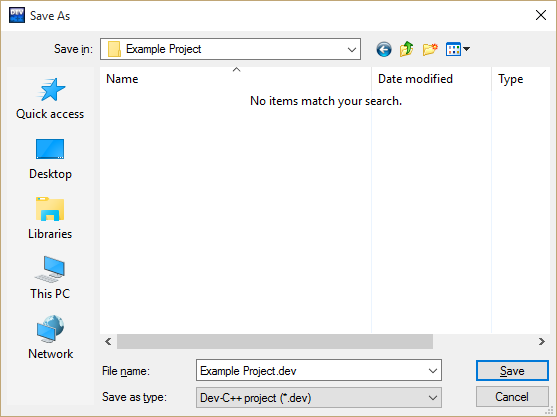
* **Creating a project In Dev C++**
* Once the install completes, create a project:



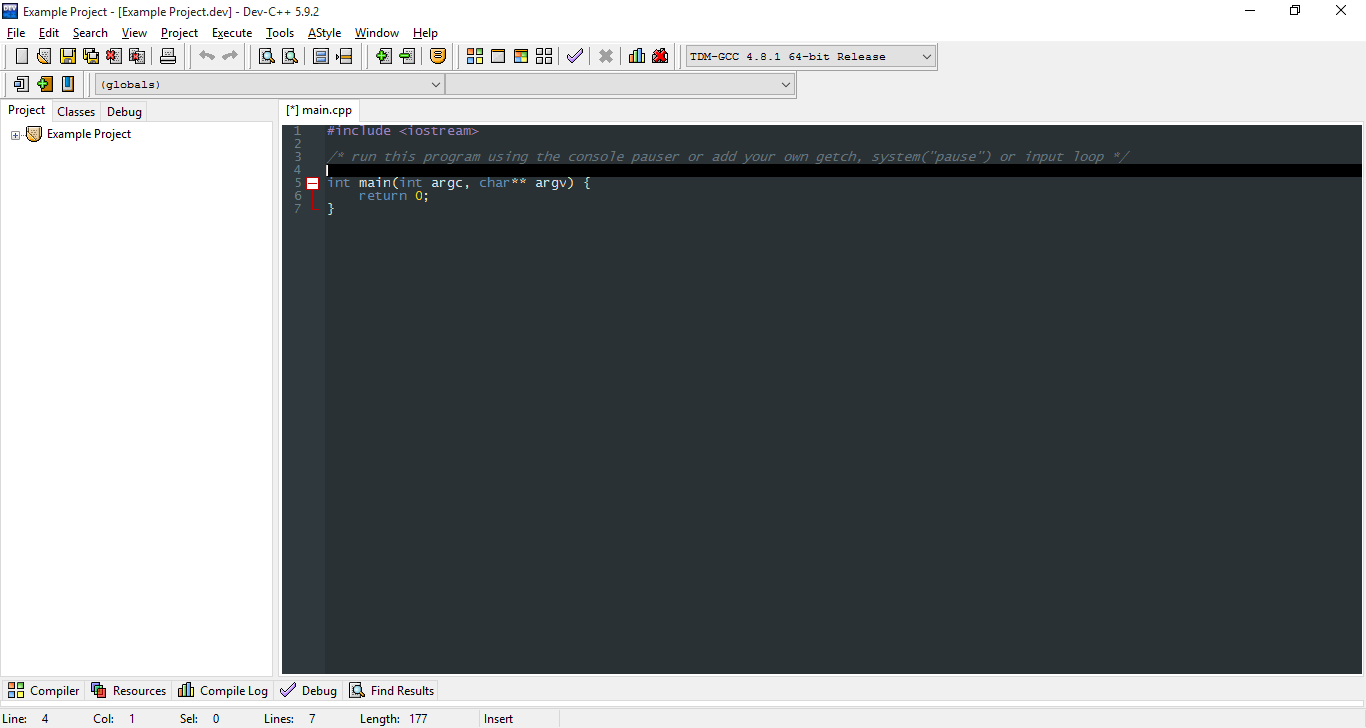
* Select “console application,” “C++ project,” and name your project:



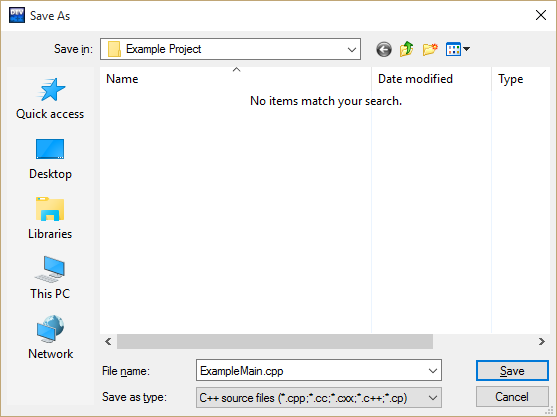
* Save your project in a new folder:



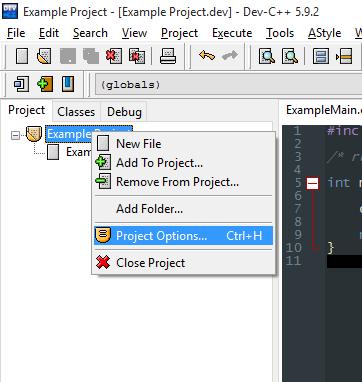
* Your project should now look like this:



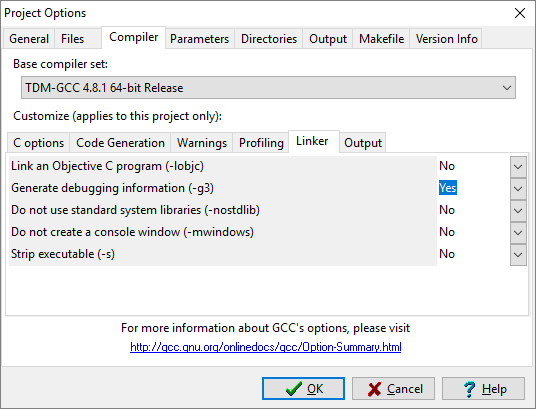
* Save the main.cpp file (to the same folder), rename it if you’d like:



* You should now have a working project. To change compiler settings:

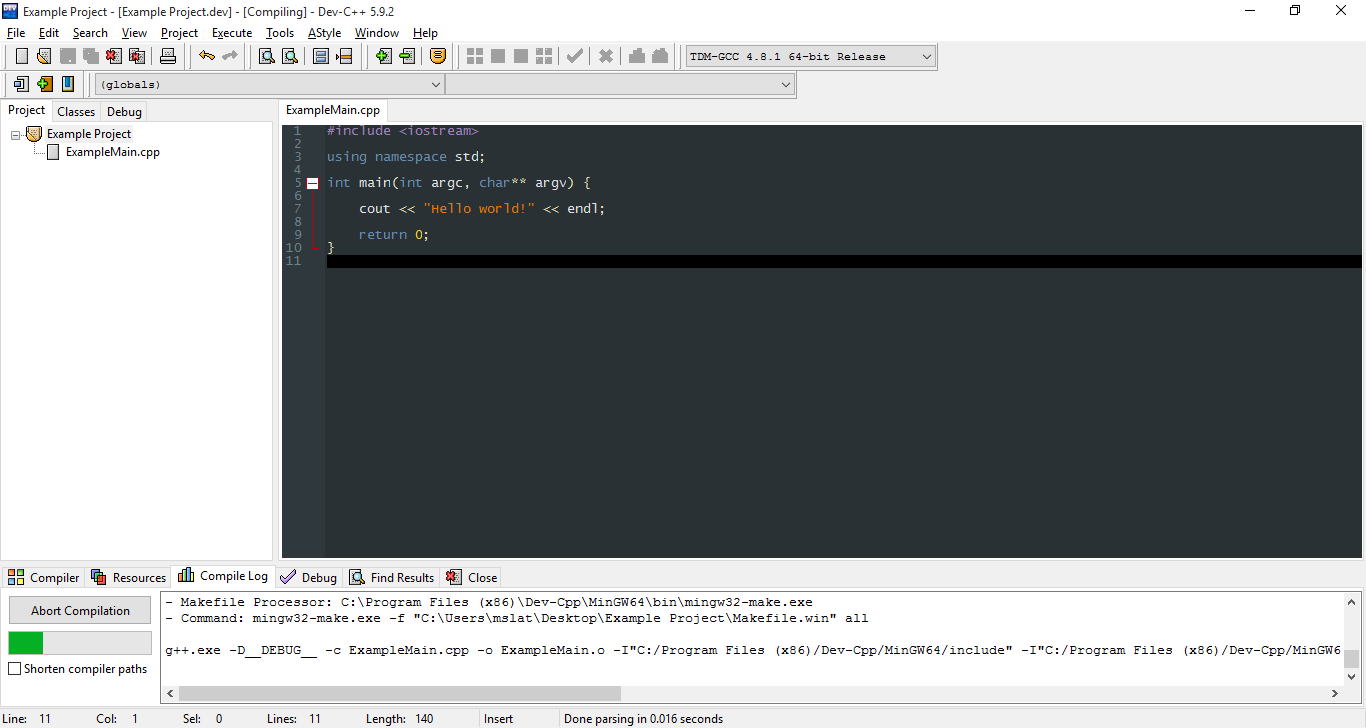


Here you can change project settings like name, type, icon, etc. To enable debugging information, enable Compiler > Linker > Generate Debugging Information (-g3). I'd also recommend enabling warnings (Compiler > Warnings > Show Most Warnings, Show More Warnings). For now, you will only be interested in the “General,” “Output,” and maybe the “Compiler” and “Directories” tabs.

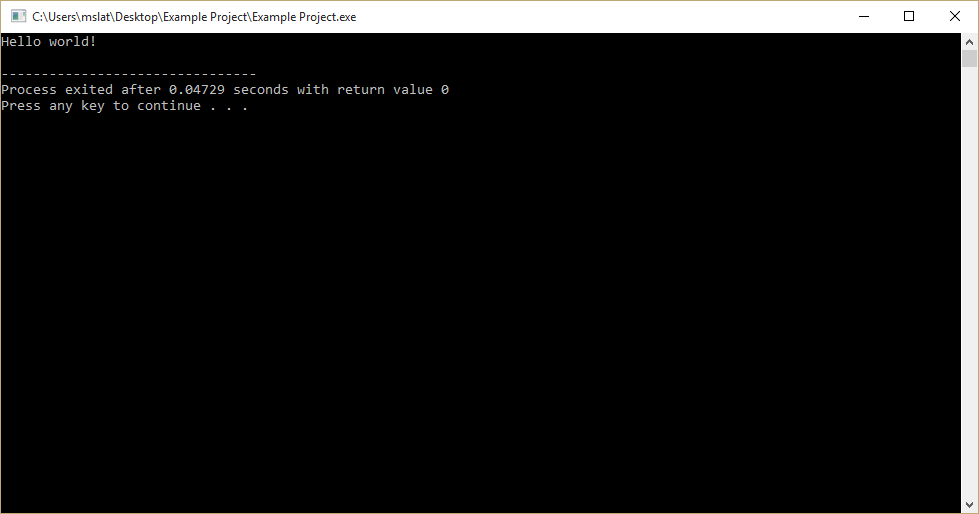


* **Development In Dev C++**

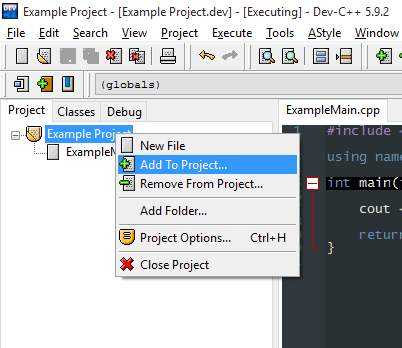
If you are satisfied with your settings, write a “hello world” program and press F11 to compile and run:



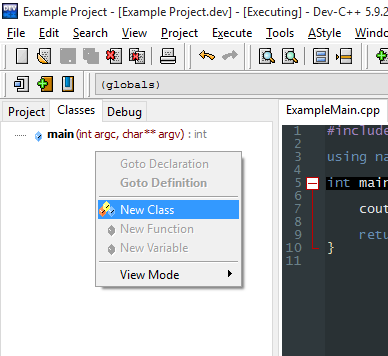
If you set up your project correctly, you should get what you expect.



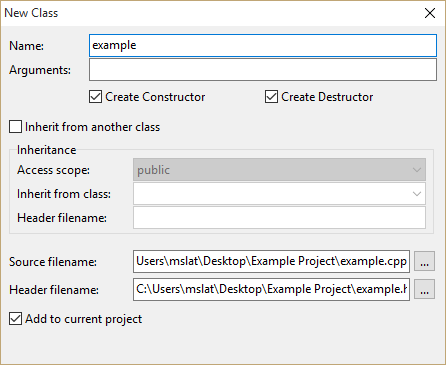
* **Adding Files**
* To add non-code files to your project, such as text or data for input:



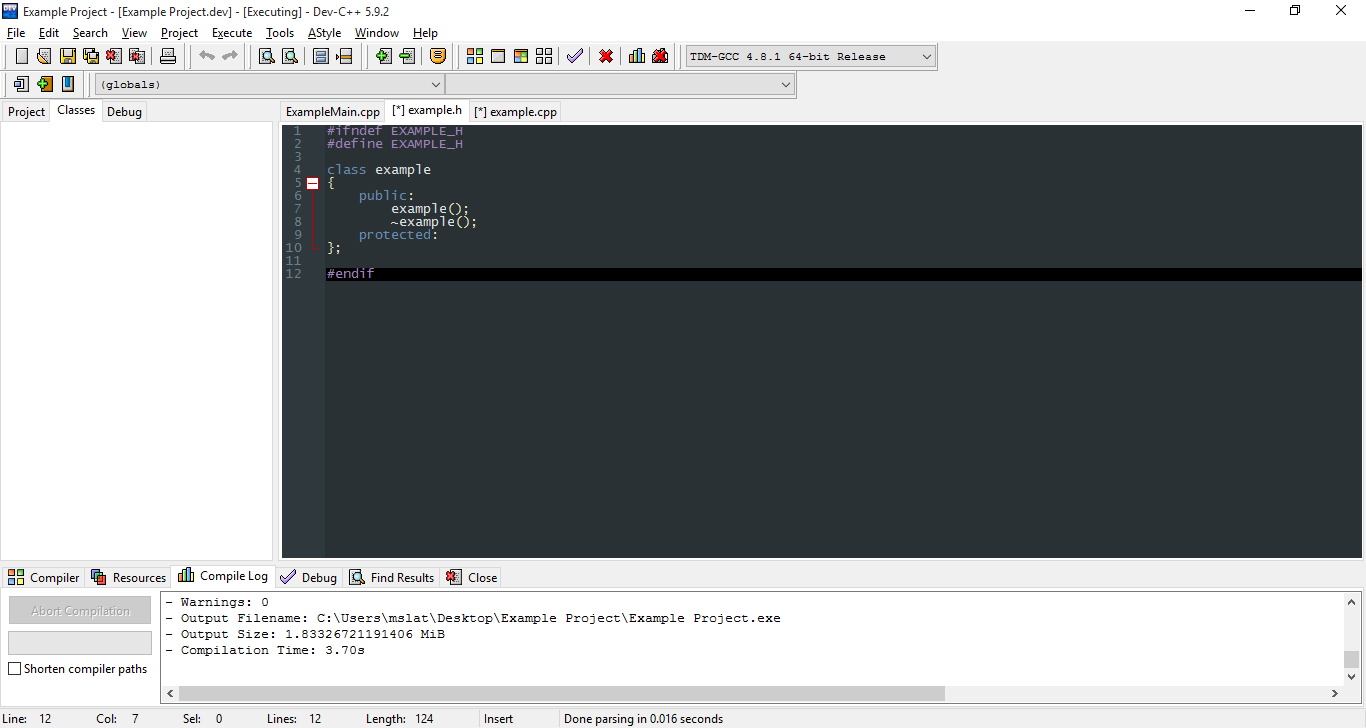
* To add a class:



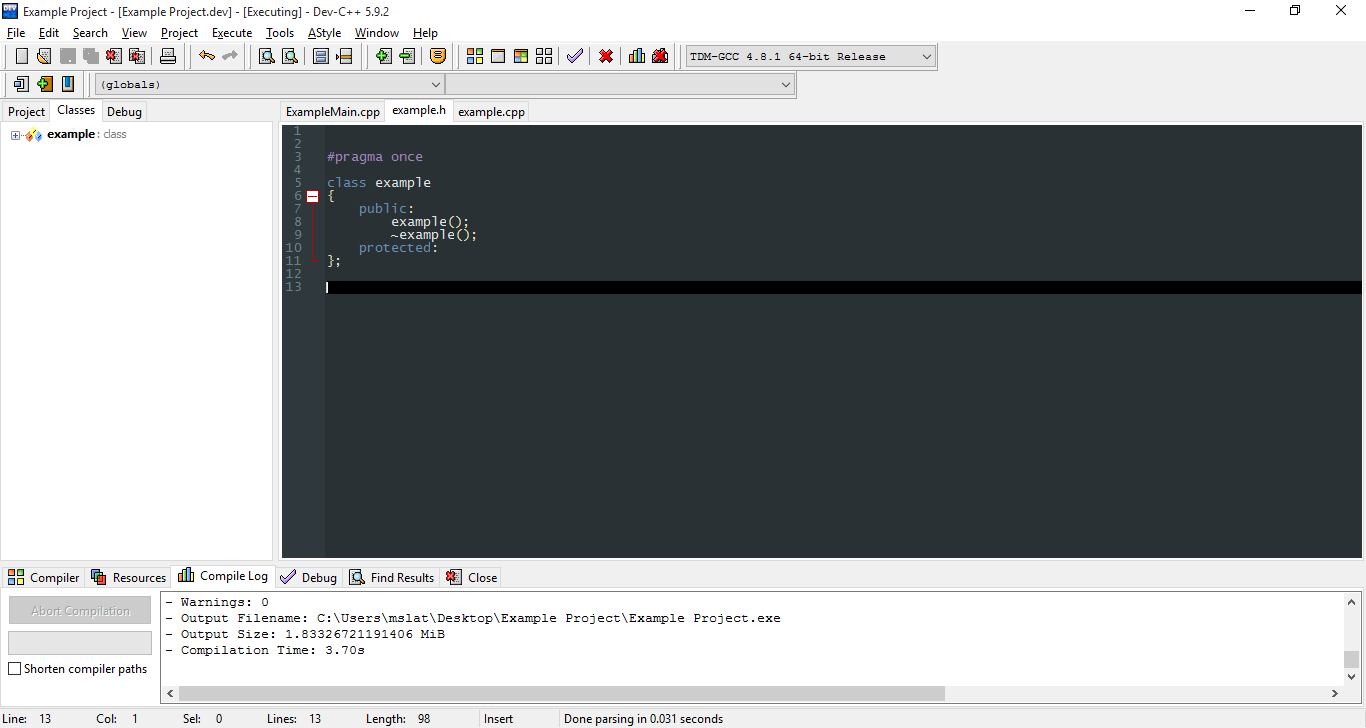
* Here, name your class and add a default constructor/destructor:



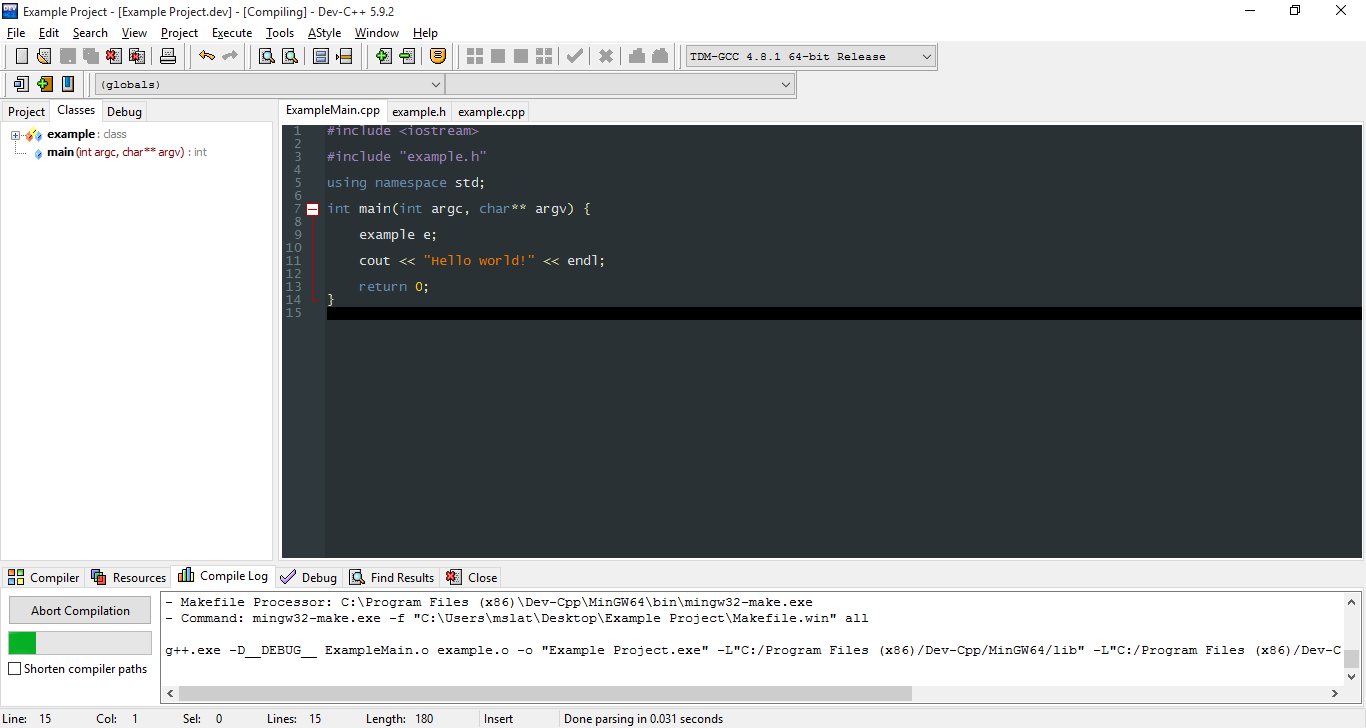
* An .h and .cpp file for your class should appear. Save them both to your project folder.



* In the .h file, you can replace all of the # statements with “#pragma once” if you’d like. This makes sure that if your .h file would be included more than once, it is only included once:



* To use your class in the main file, simply include your new .h file, using quotes. Try compiling this, to make sure everything is working. To add more classes, simply repeat this process.



* Finally, you are finished! Remember to save your project settings when you close Dev C++. Your project folder should now have several unfamiliar files. The .dev file is your project, and is what you will open when you want to work on your project.
* The .layout , .o, and make file files are all by products of compiling with multiple files, so leave them be and you should be good.

