

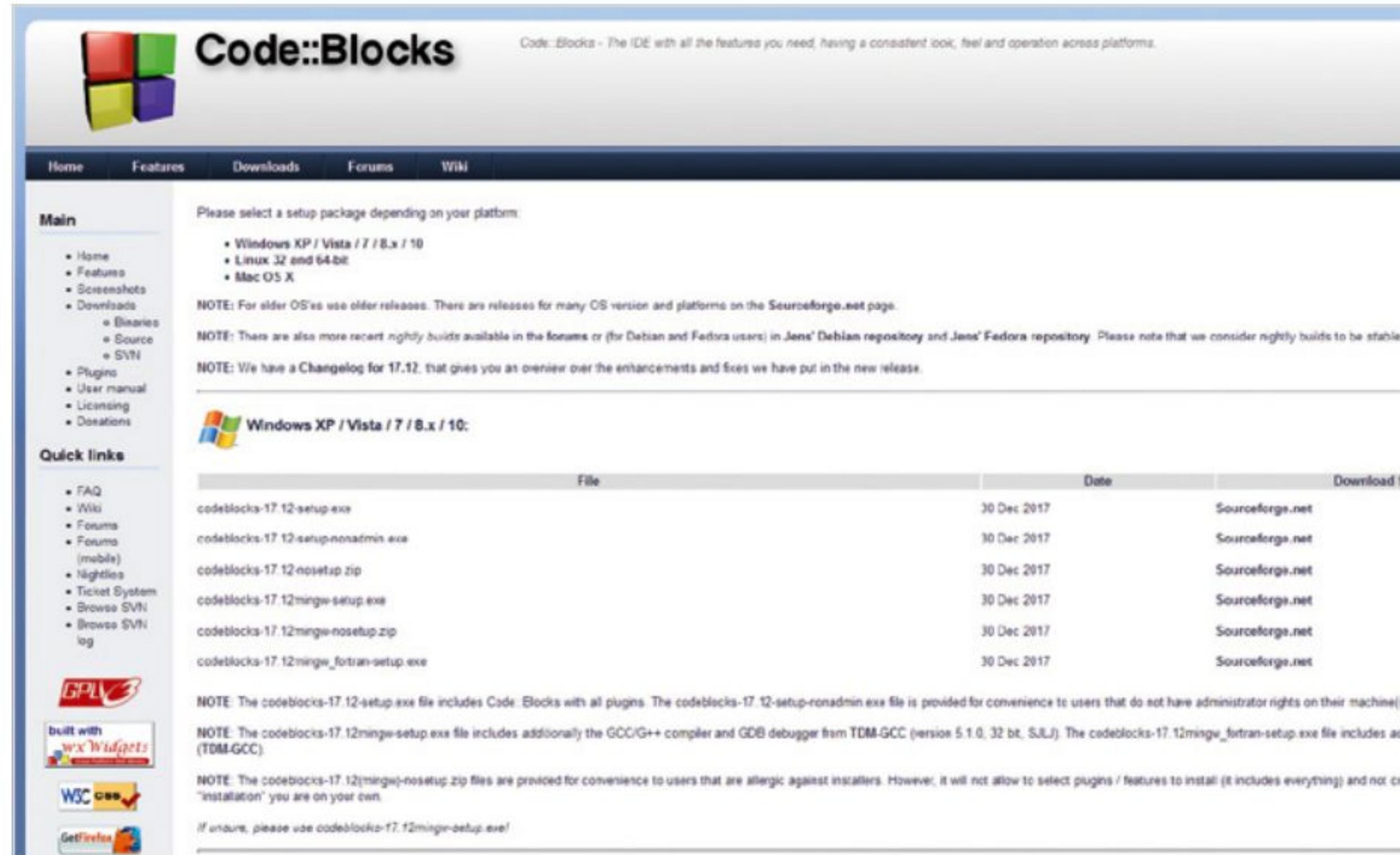
How to Set Up C++ in Windows

Windows users have a wealth of choice when it comes to programming in C++. There are loads of IDEs and compilers available, including Visual Studio from Microsoft. However, in our opinion, the best C++ IDE to begin with is Code::Blocks.

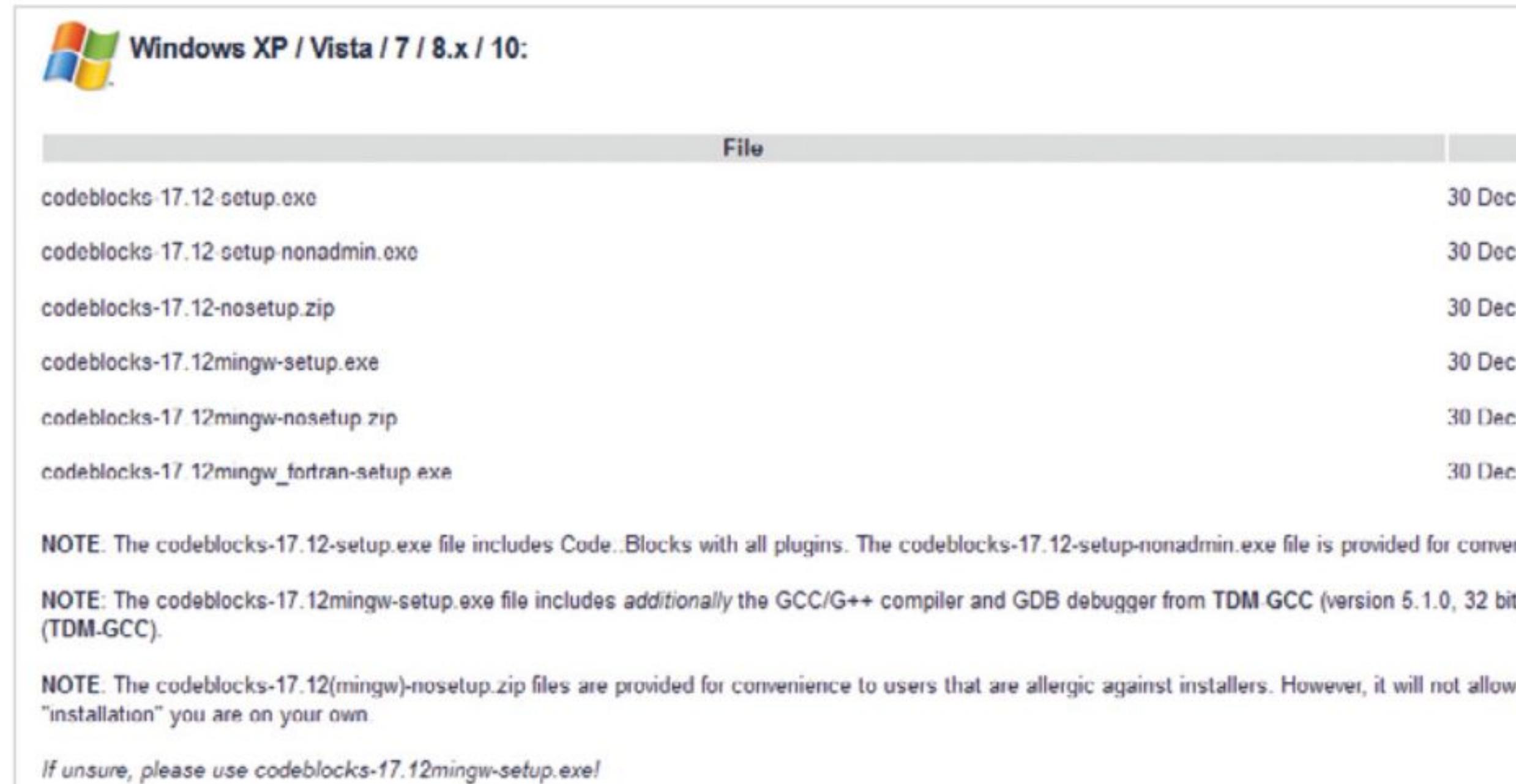
CODE::BLOCKS

Code::Blocks is a free C++, C and Fortran IDE that is feature rich and easily extendible with plugins. It's easy to use, comes with a compiler and has a vibrant community behind it too.

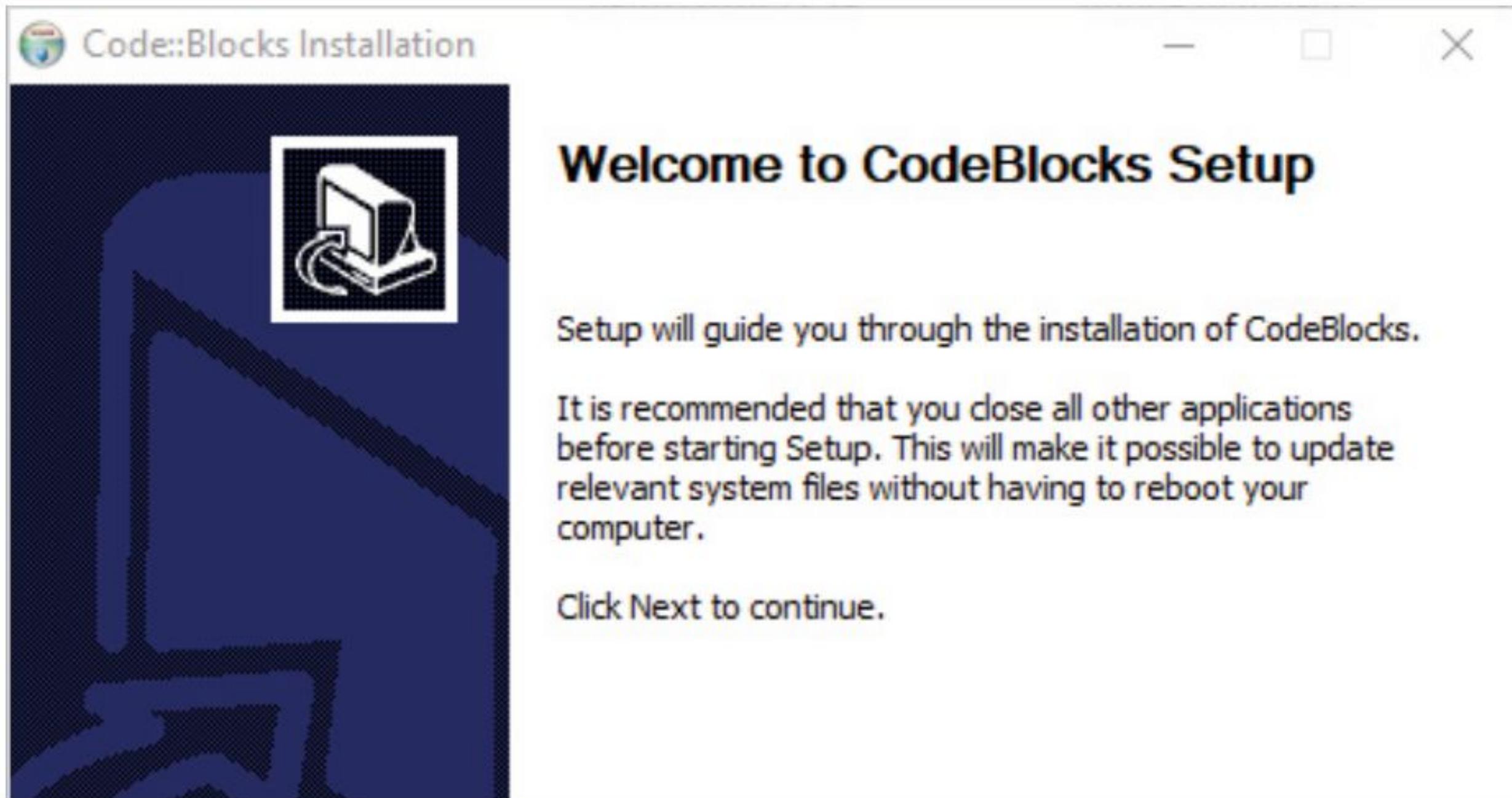
STEP 1 Start by visiting the Code::Blocks download site, at www.codeblocks.org/downloads. From there, click on the 'Download the binary releases' link to be taken to the latest downloadable version for Windows.



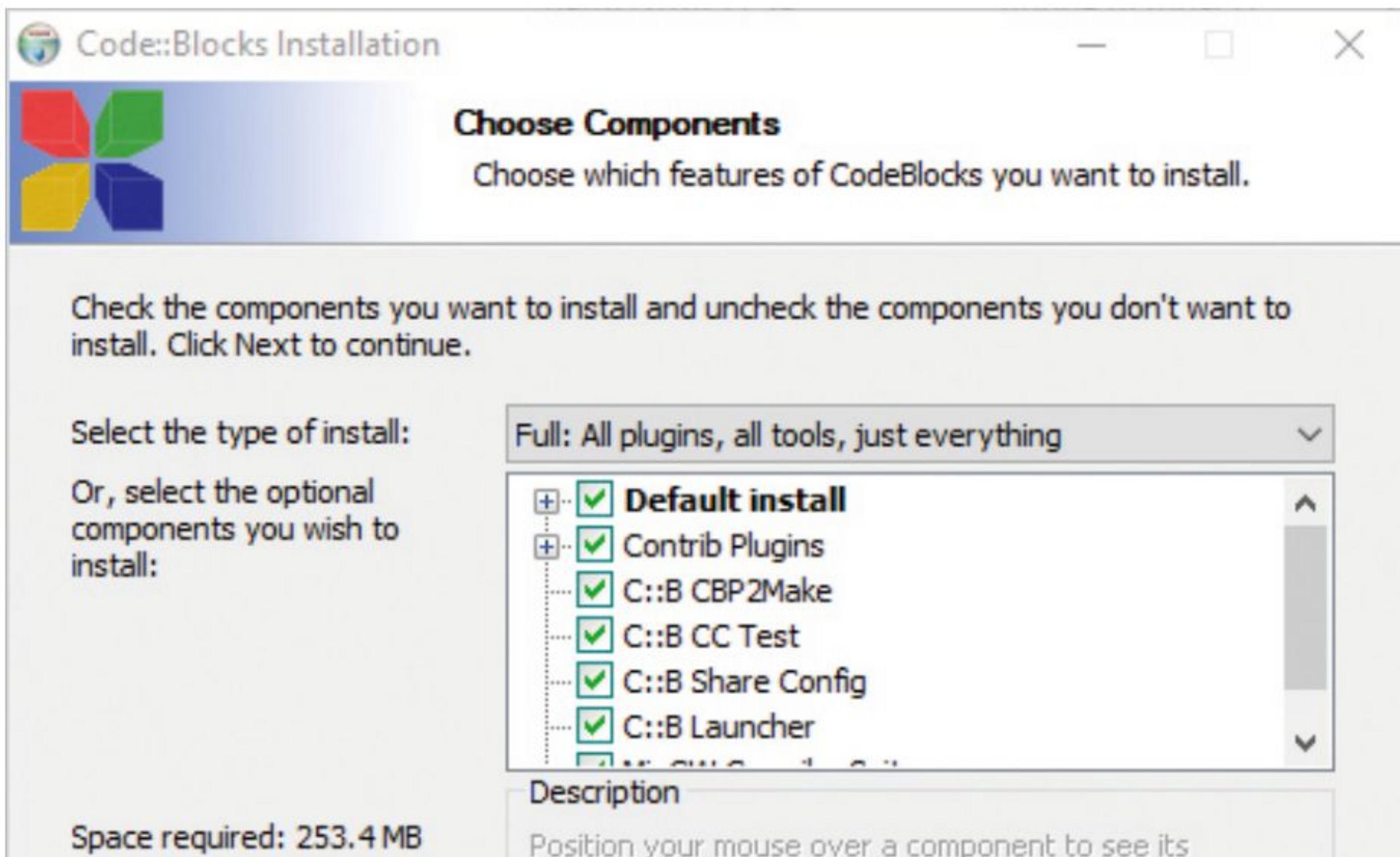
STEP 2 There you can see, there are several Windows versions available. The one you want to download has mingw-setup.exe at the end of the current version number. At the time of writing this is: codeblocks-17.12mingw-setup.exe. The difference is that the mingw-setup version includes a C++ compiler and debugger from TDM-GCC (a compiler suite).



STEP 3 When you've located the file, click on the Sourceforge.net link at the end of the line and a download notification window appears; click on Save File to start the download and save the executable to your PC. Locate the downloaded Code::Blocks installer and double-click to start. Follow the on-screen instructions to begin the installation.

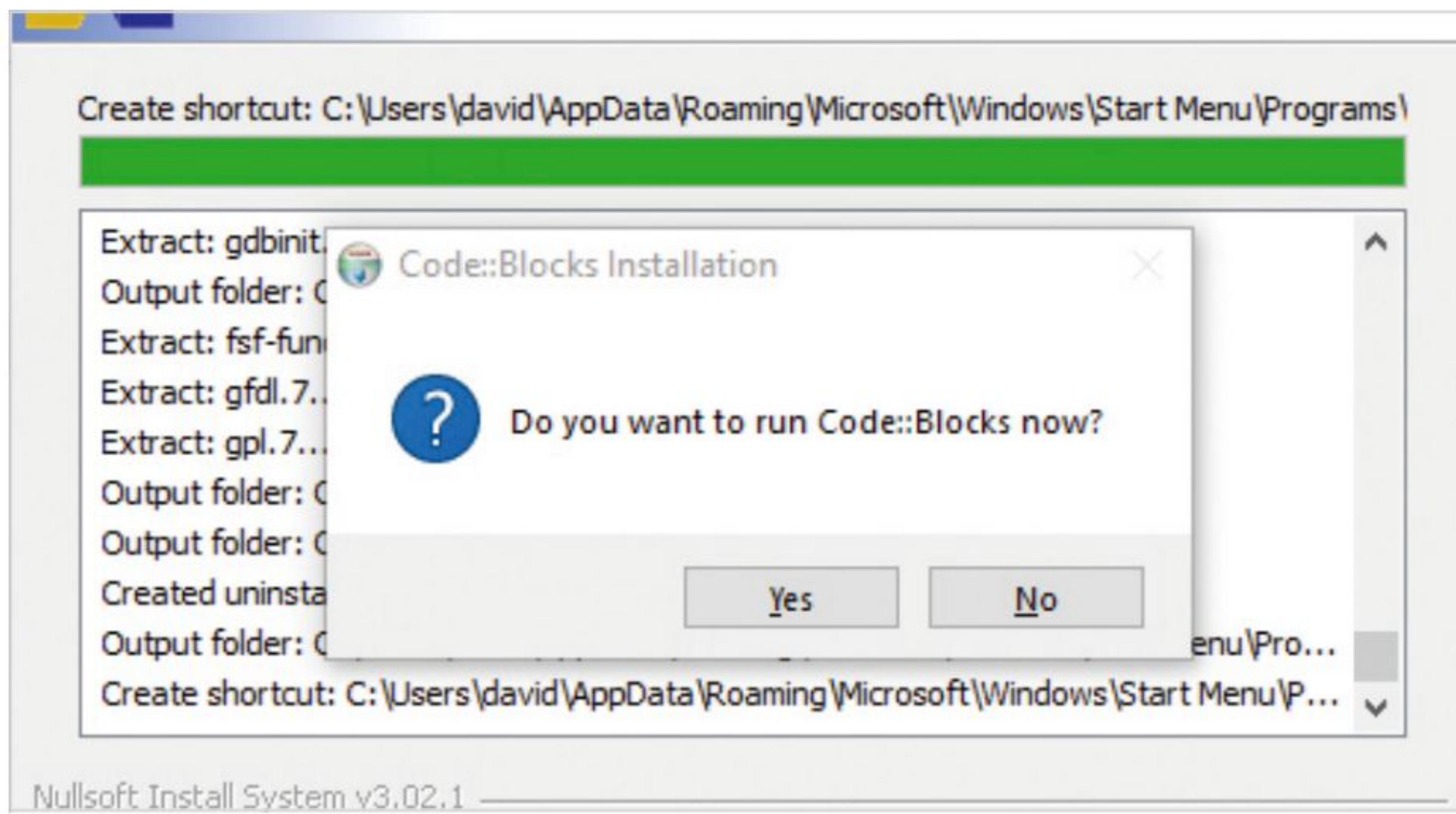


STEP 4 Once you agree to the licencing terms, a choice of installation options becomes available. You can opt for a smaller install, missing out on some of the components but we recommend that you opt for the Full option, as default.

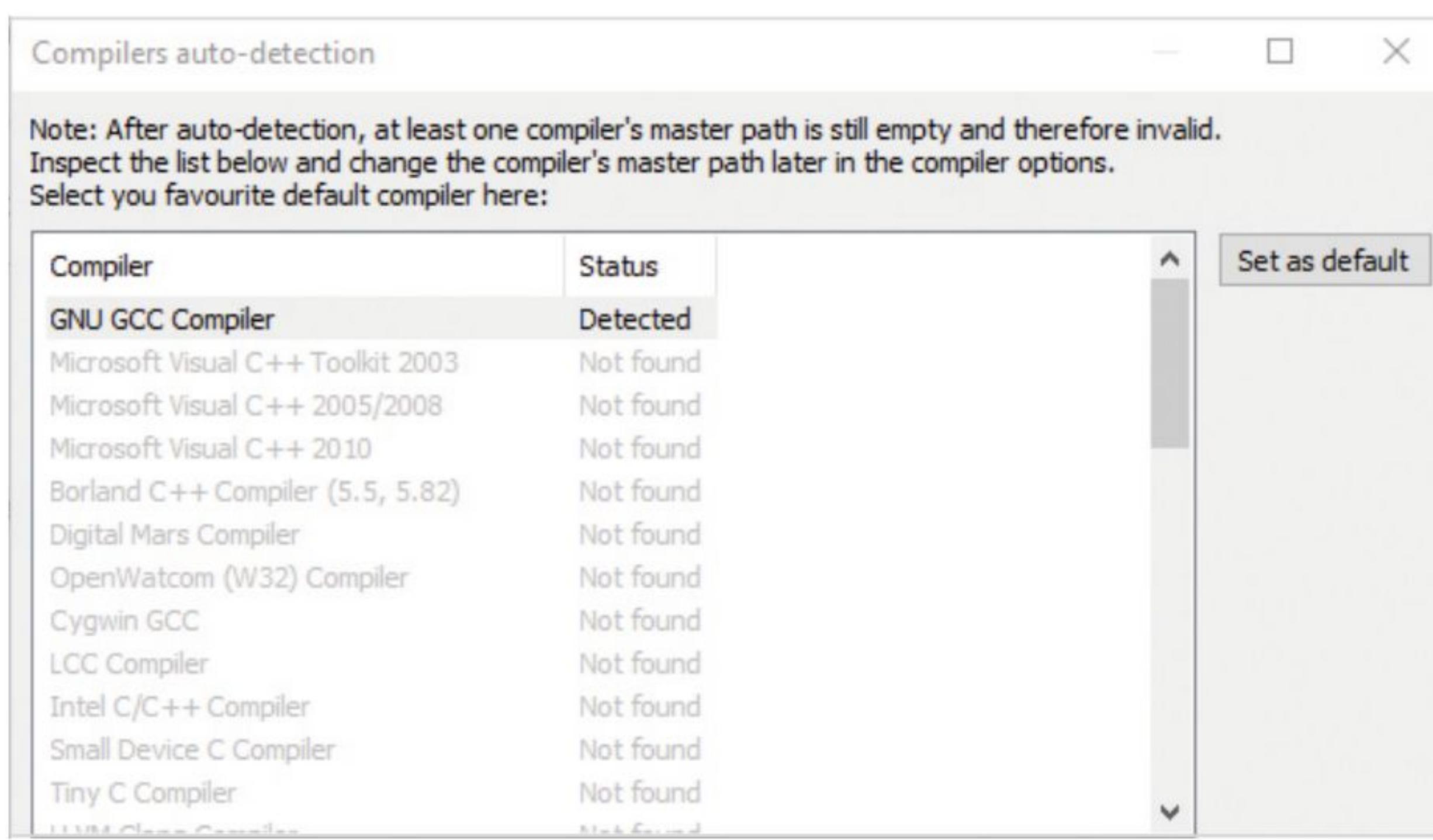




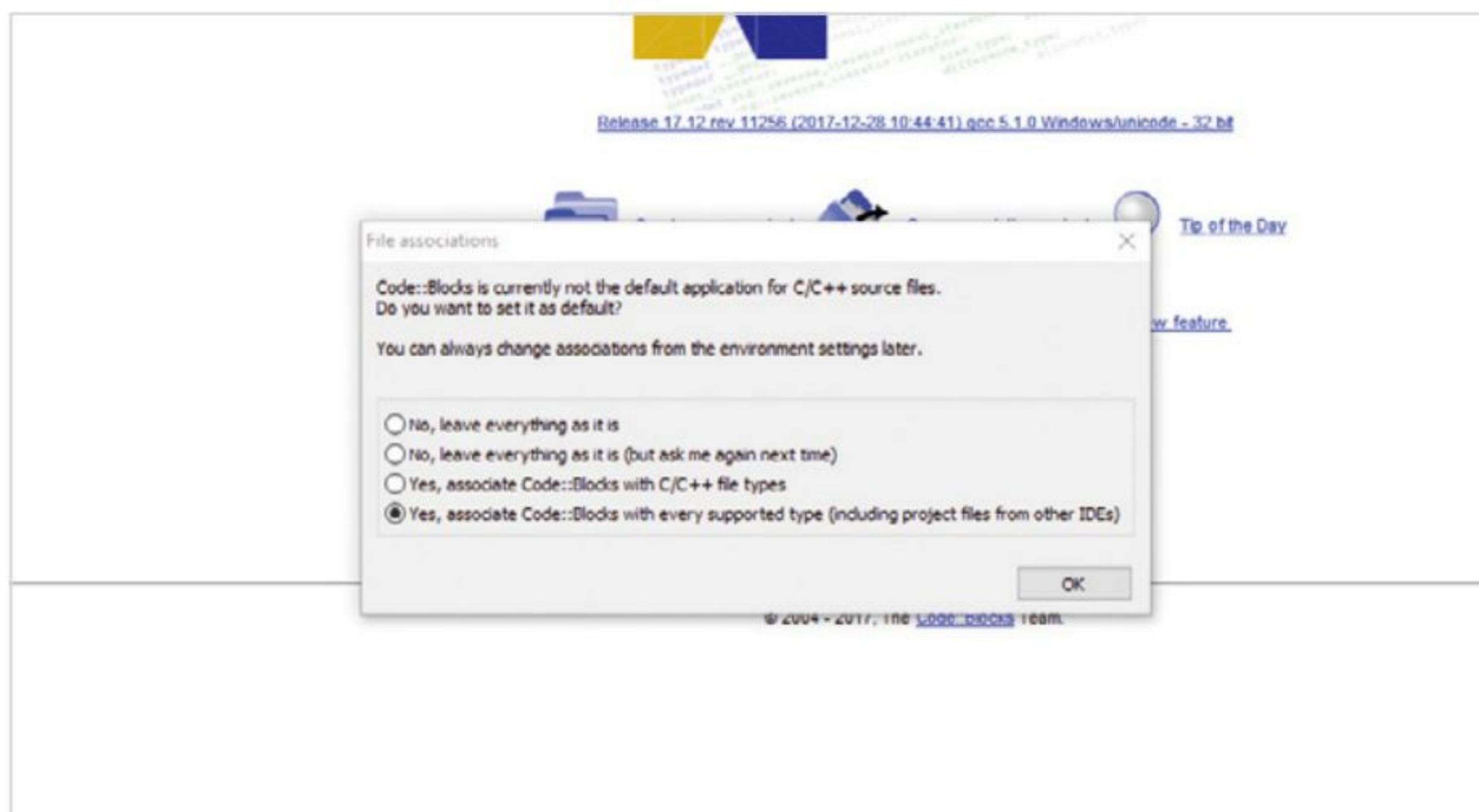
STEP 5 Next choose an install location for the Code::Blocks files. It's your choice but the default is generally sufficient (unless you have any special requirements of course). When you click Next, the install begins; when it's finished a notification pops up asking you if you want start Code::Blocks now, so click Yes.



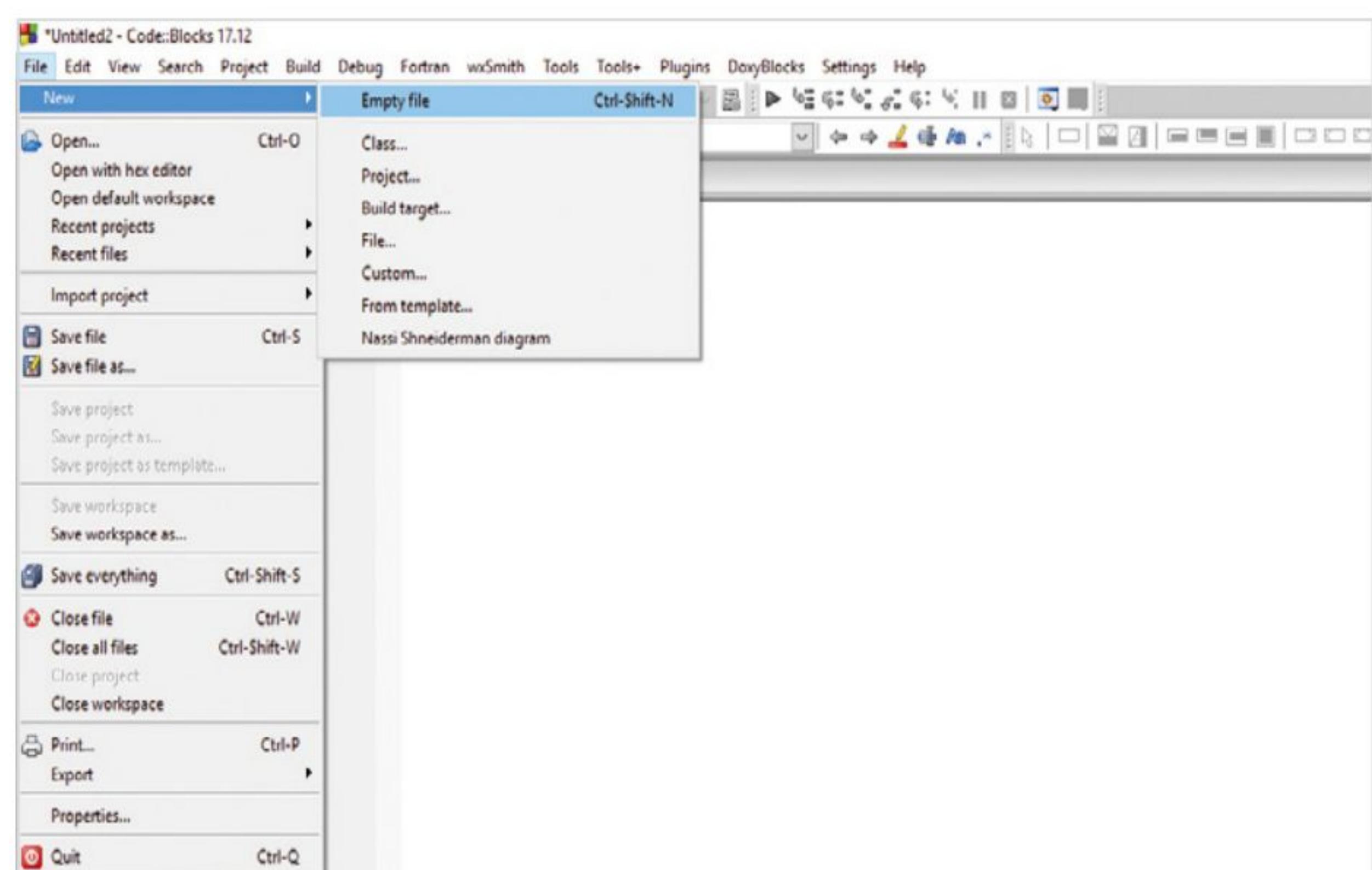
STEP 6 The first time Code::Blocks loads it runs an autodetect for any C++ compilers you may already have installed on your system. If you don't have any, click on the first detected option: GNU GCC Compiler and click the Default button to set it as the system's C++ compiler. Click OK when you're ready to continue.



STEP 7 The program starts and another message appears informing you that Code::Blocks is currently not the default application for C++ files. You have two options, to leave everything as it is or allow Code::Blocks to associate all C++ file types. Again, we would recommend you opt for the last choice, to associate Code::Blocks with every supported file type.



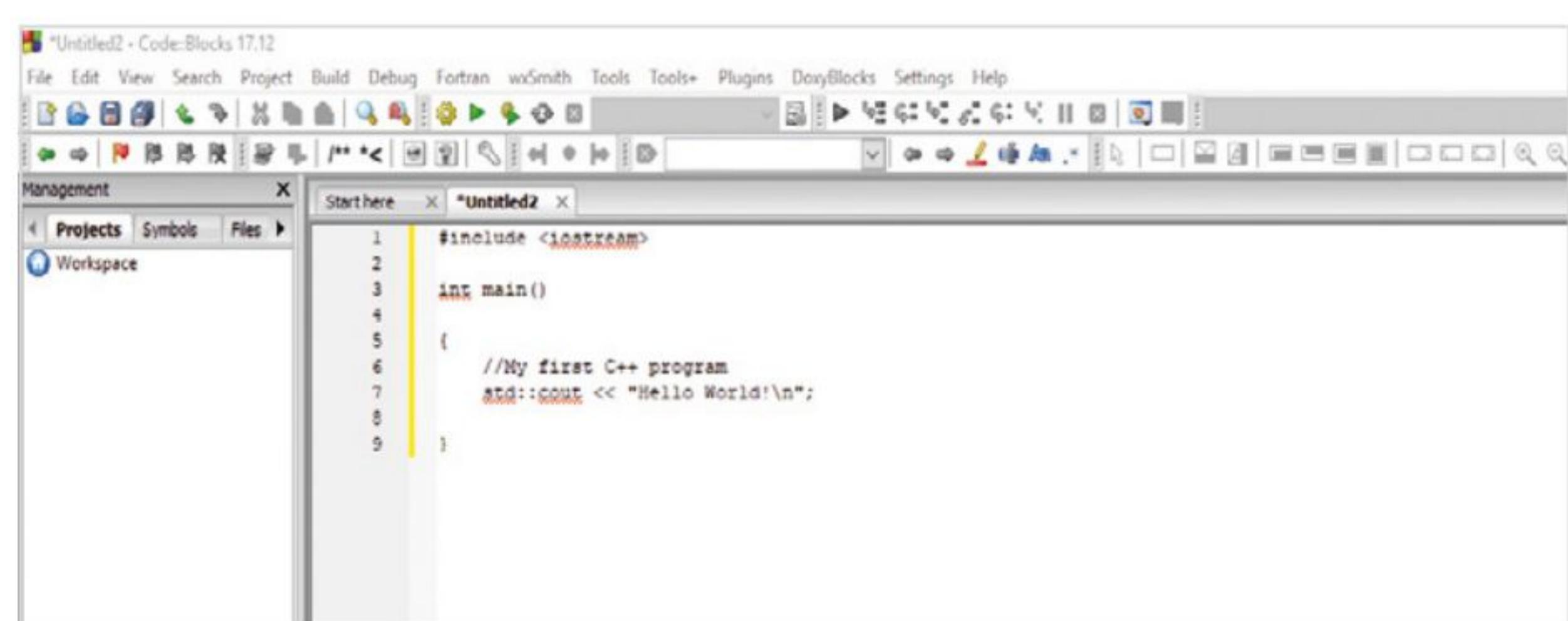
STEP 8 There's a lot you can do in Code::Blocks, so you need to dig in and find a good C++ tutorial to help you get the most from it. However, to begin with, click on File > New > Empty File. This creates a new, blank window for you to type in.



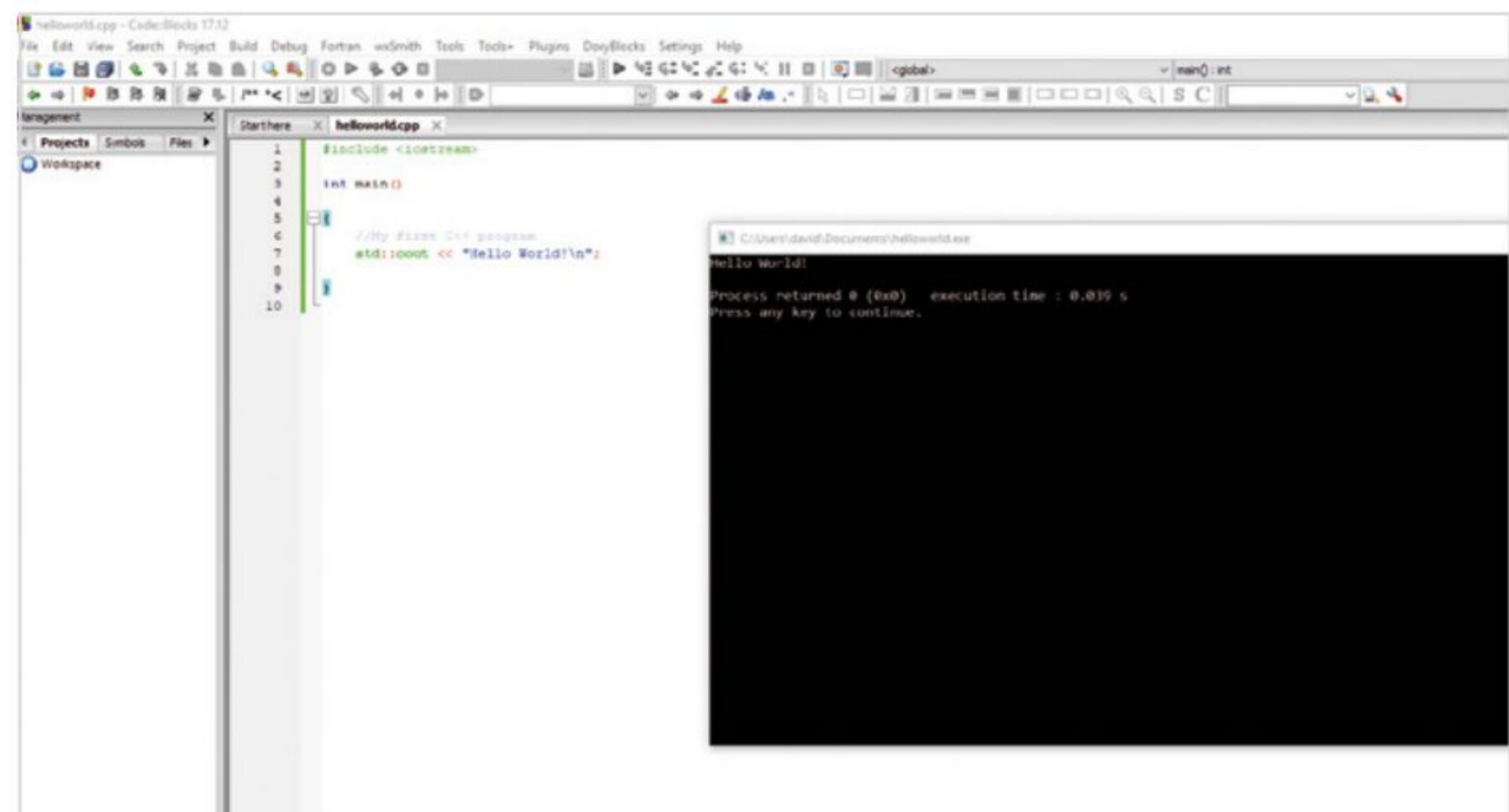
STEP 9 In the new window, enter the following:

```
#include <iostream>
Int main()
{
//My first C++ program
Std::cout << "Hello World!\n";
}
```

Notice how Code::Blocks auto-inserts the braces and speech quotes.



STEP 10 Click File > Save as and save the code with a .cpp extension (helloworld.cpp, for example). Code::Blocks changes the view to colour code according to C++ standards. To execute the code, click on the Build and Run icon along the top of the screen. It's a green play icon together with a yellow cog.





How to Set Up C++ on a Mac

To start C++ coding on a Mac you need to install Apple's Xcode. This is a free, full featured IDE that's designed to create native Apple apps. However, it can also be used to create C++ code relatively easily.

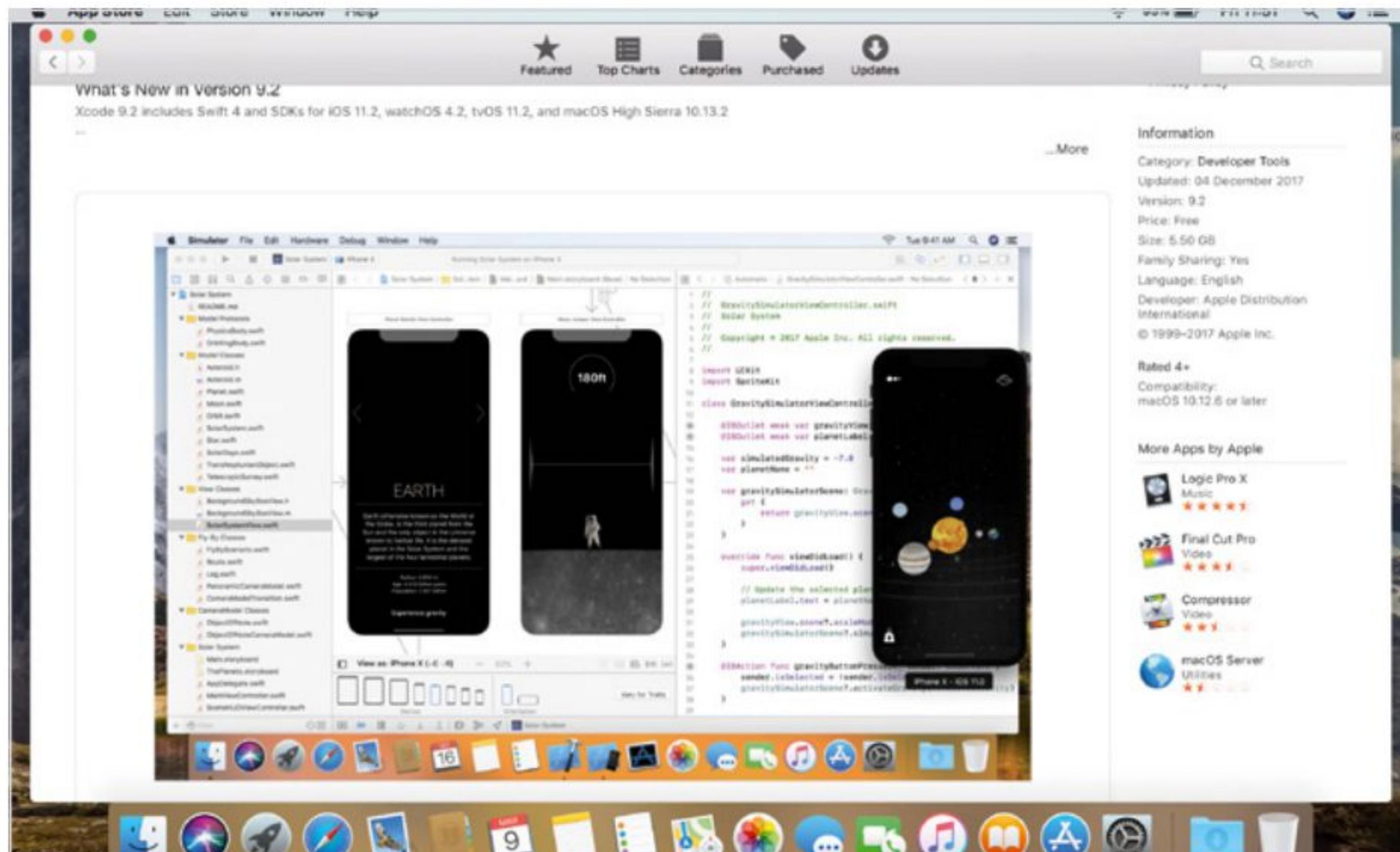
XCODE

Apple's Xcode is primarily designed for users to develop apps for macOS, iOS, iPadOS, tvOS and watchOS applications in Swift or Objective-C, but you can use it for C++ too.

- STEP 1** Start by opening the App Store on your Mac, Apple Menu > App Store. In the Search box enter Xcode and press Return. There will be many suggestions filling the App Store window but it's the first option, Xcode, that you need to click on.



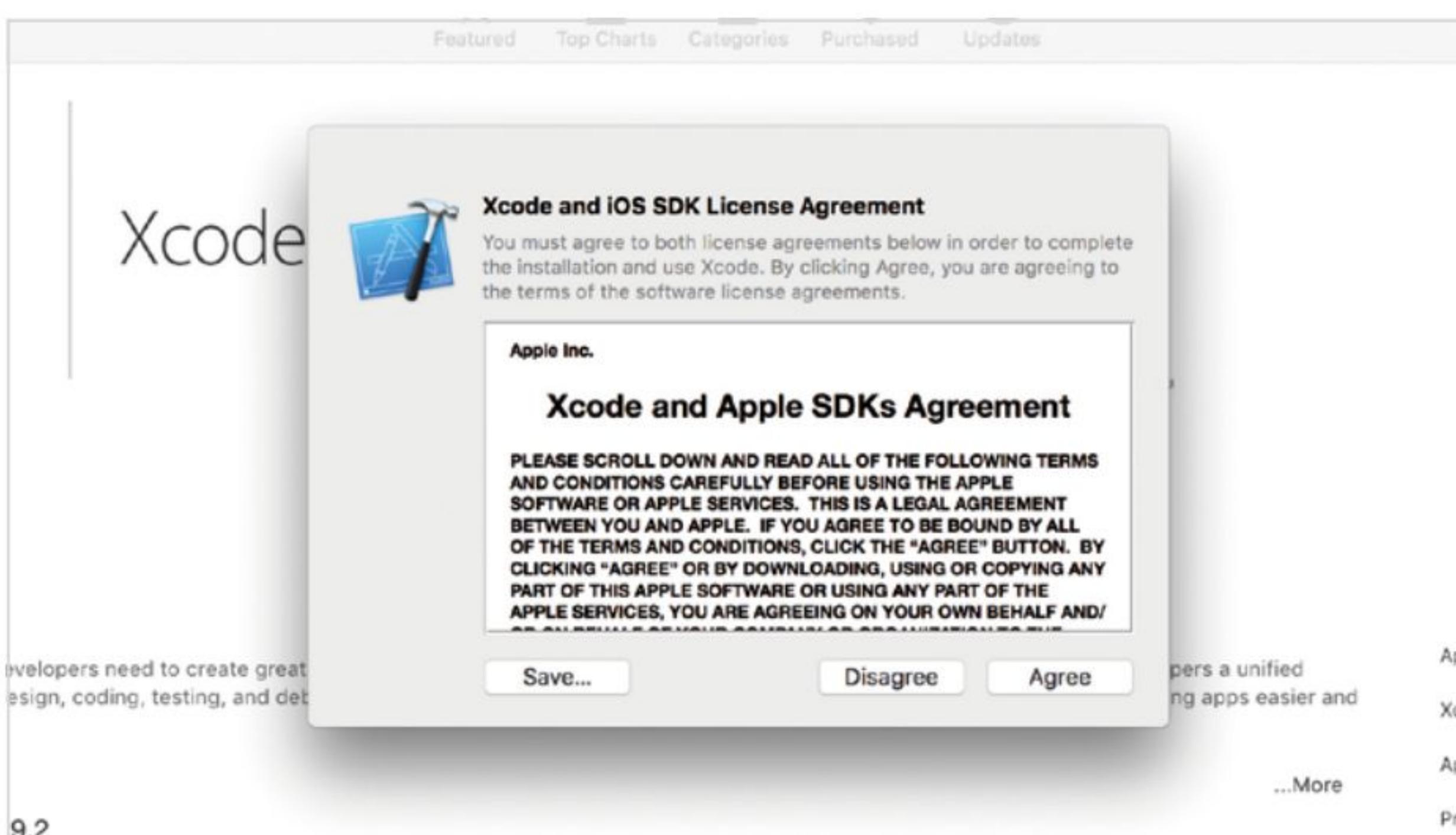
- STEP 2** Take a moment to browse through the app's information, including the compatibility to ensure you have the correct version of macOS. Xcode requires macOS 10.12.6 or later to install and work.



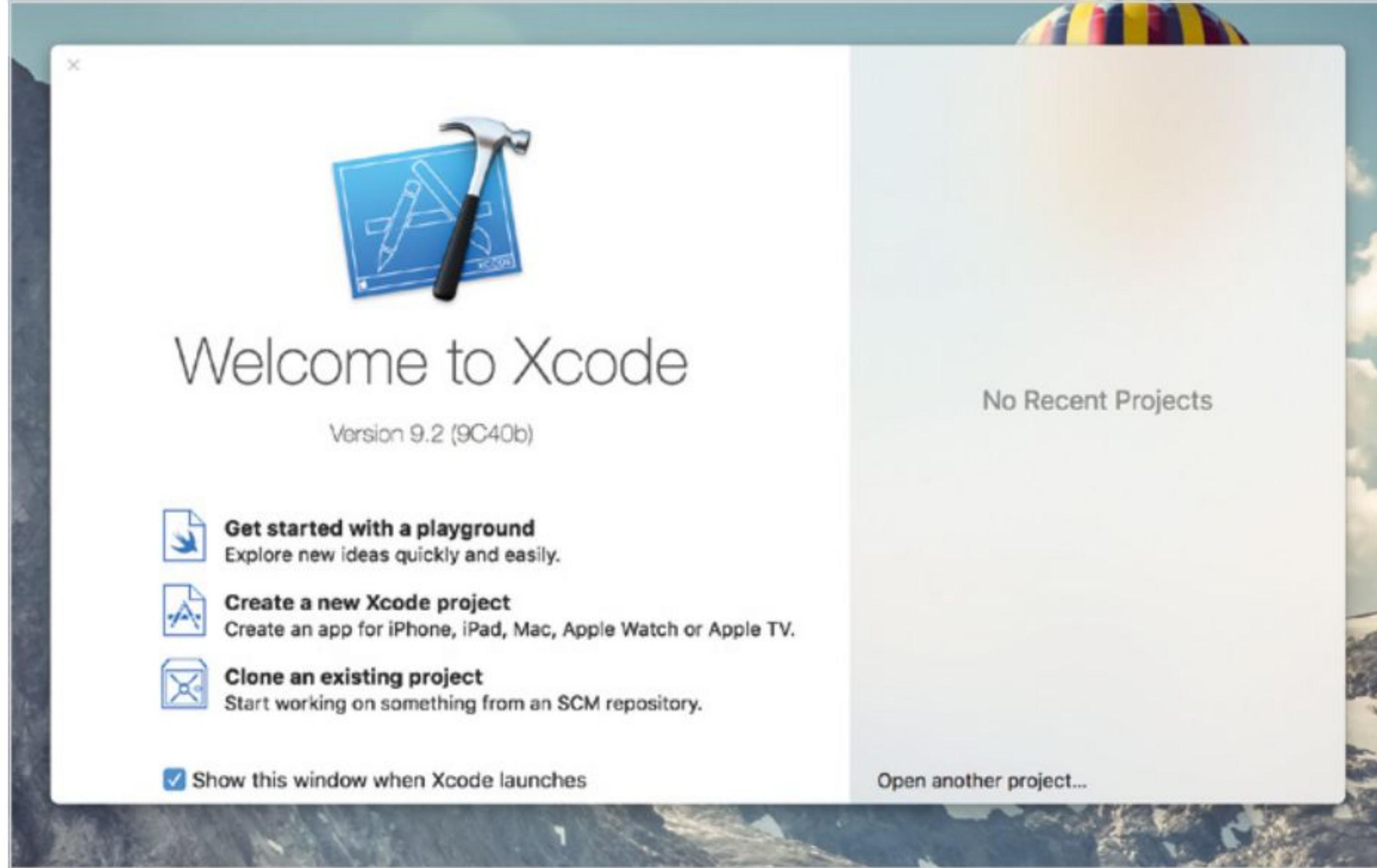
- STEP 3** When you're ready, click on the Get button which then turns into 'Install App'. Enter your Apple ID and Xcode begins to download and install. It may take some time depending on the speed of your Internet connection.



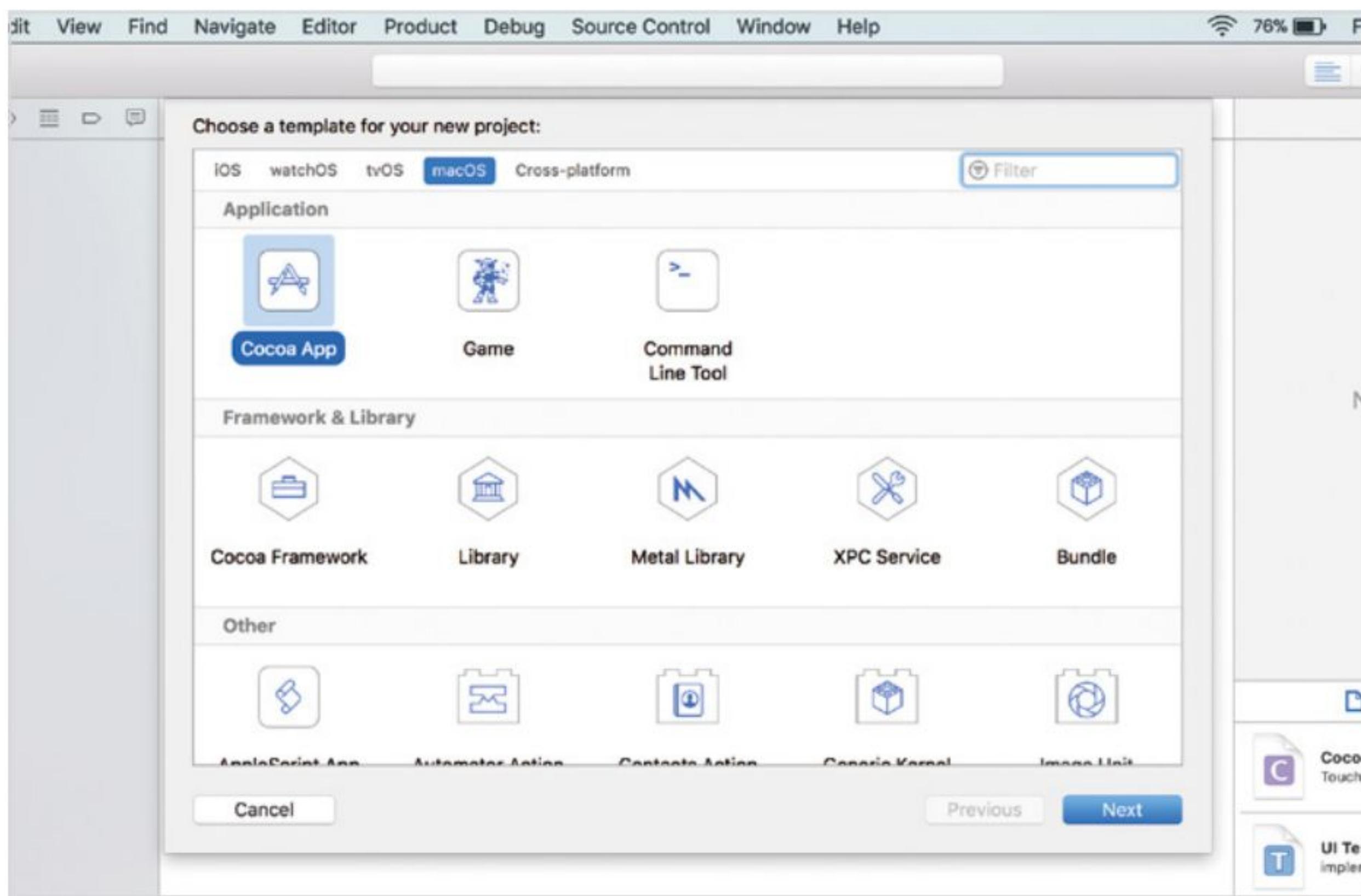
- STEP 4** When the installation is complete, click on the Open button to launch Xcode. Click Agree to the licence terms and enter your password to allow Xcode to make changes to the system. When you've done that, Xcode begins to install additional components.



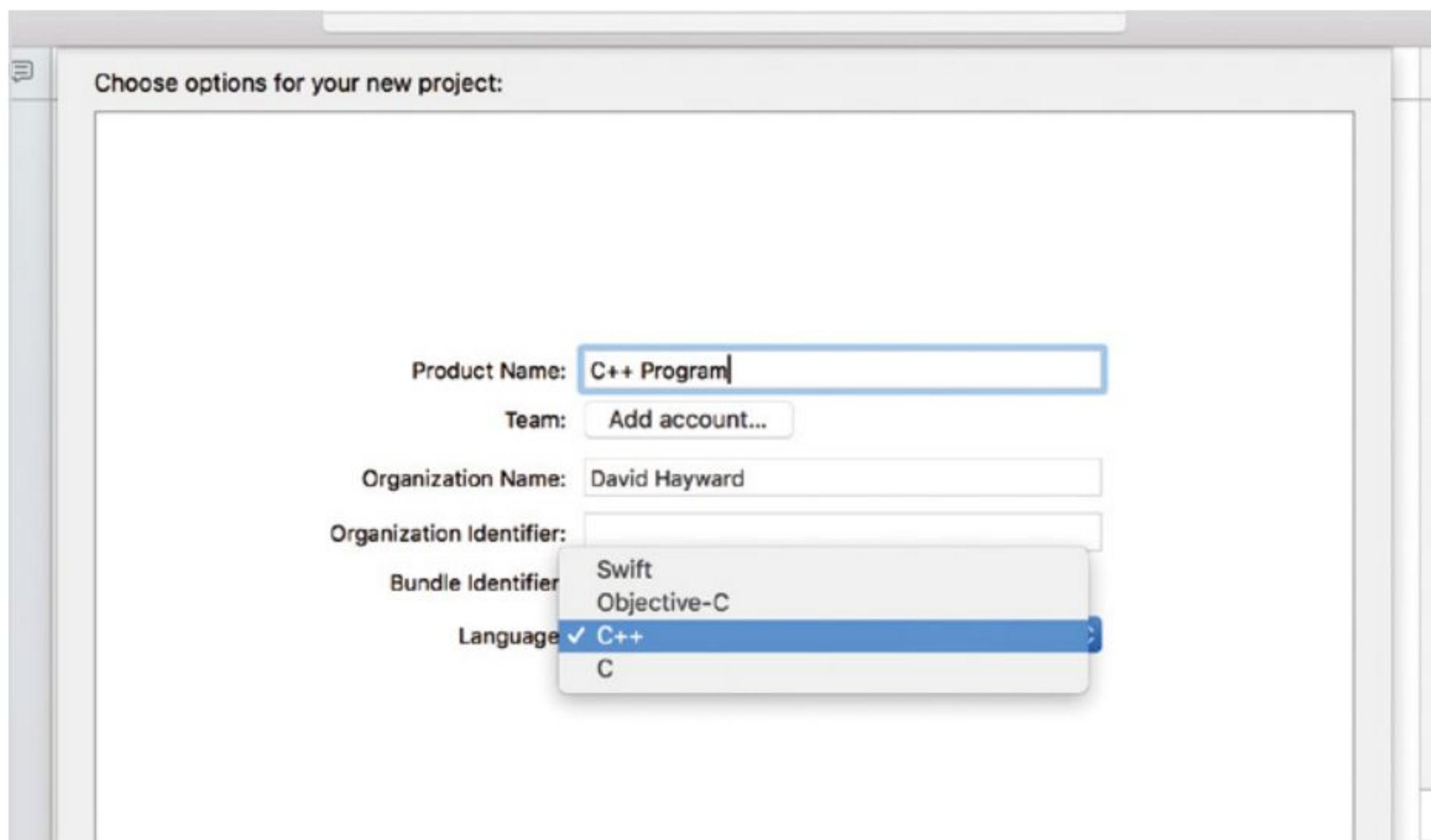
STEP 5 With everything now installed, including the additional components, Xcode launches, displaying the version number along with three choices and any recent projects that you've worked on; although for a fresh install, this shows blank.



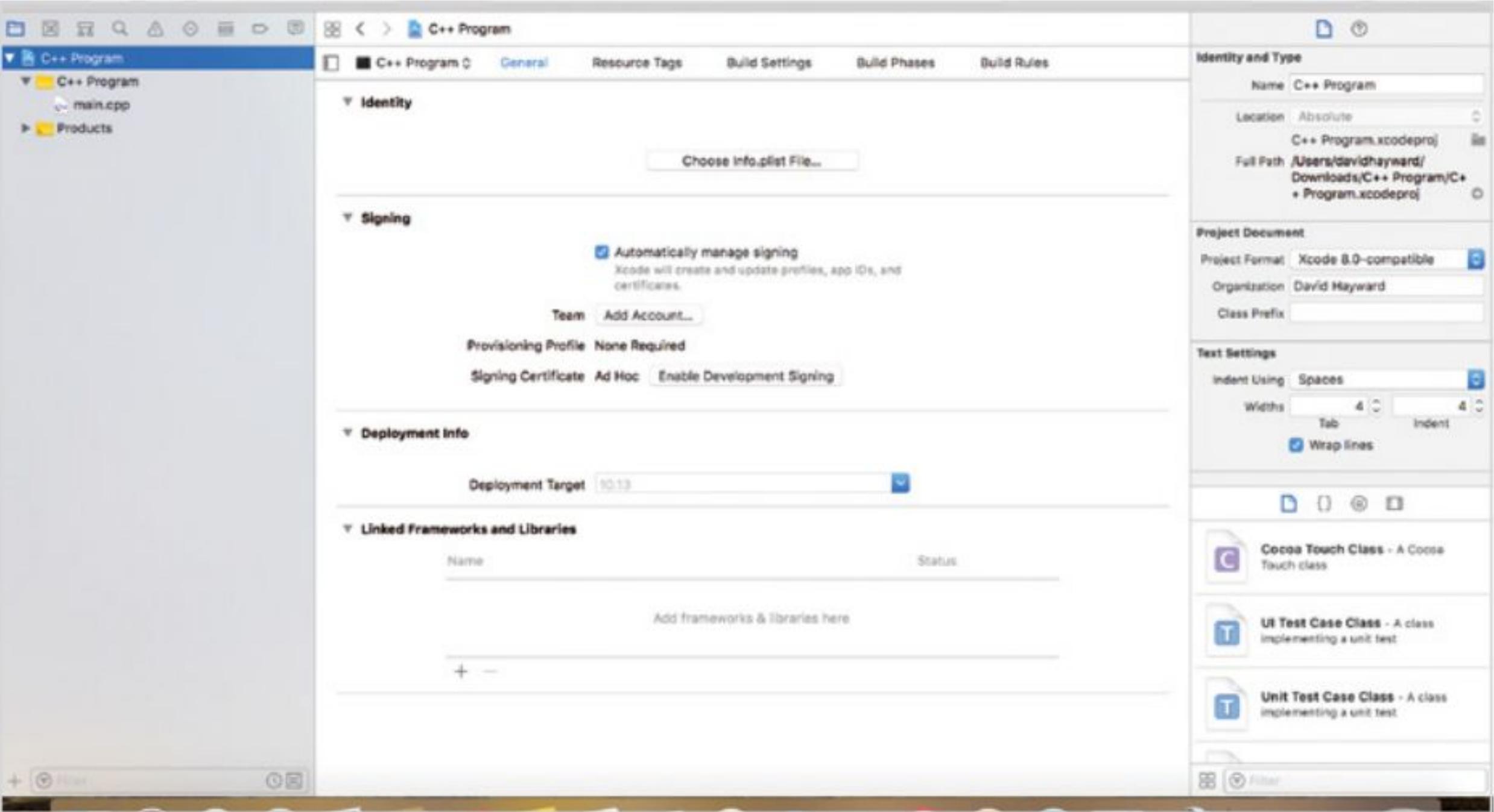
STEP 6 Start by clicking on Create New Xcode Project; this opens a template window from which to choose the platform you're developing code for. Click the macOS tab, then the Command Line Tool option and finally, Next to continue.



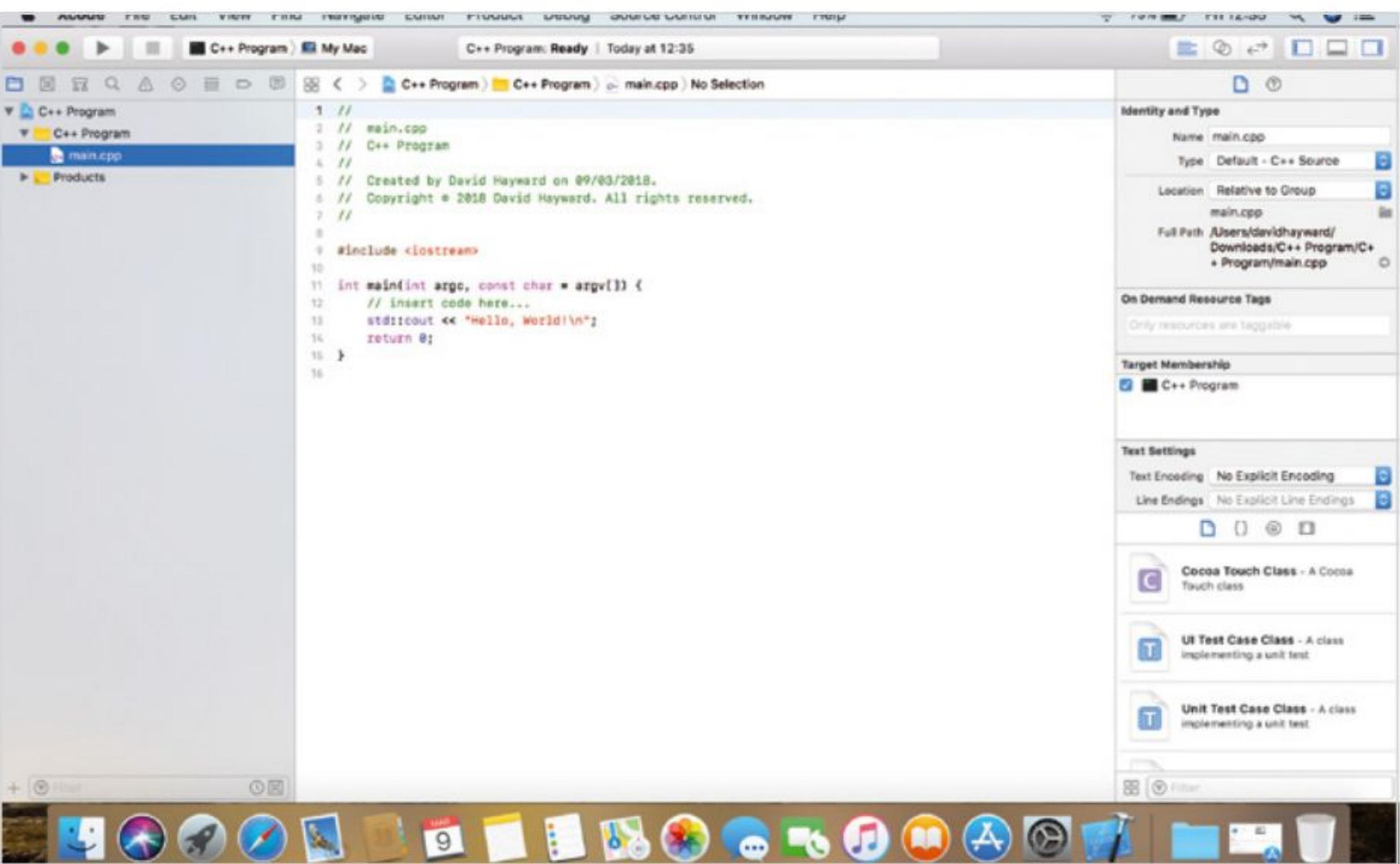
STEP 7 Fill in the various fields but ensure that the Language option at the bottom is set to C++. Simply choose it from the drop-down list. When you've filled in the fields, and made sure that C++ is the chosen language, click on the Next button to continue.



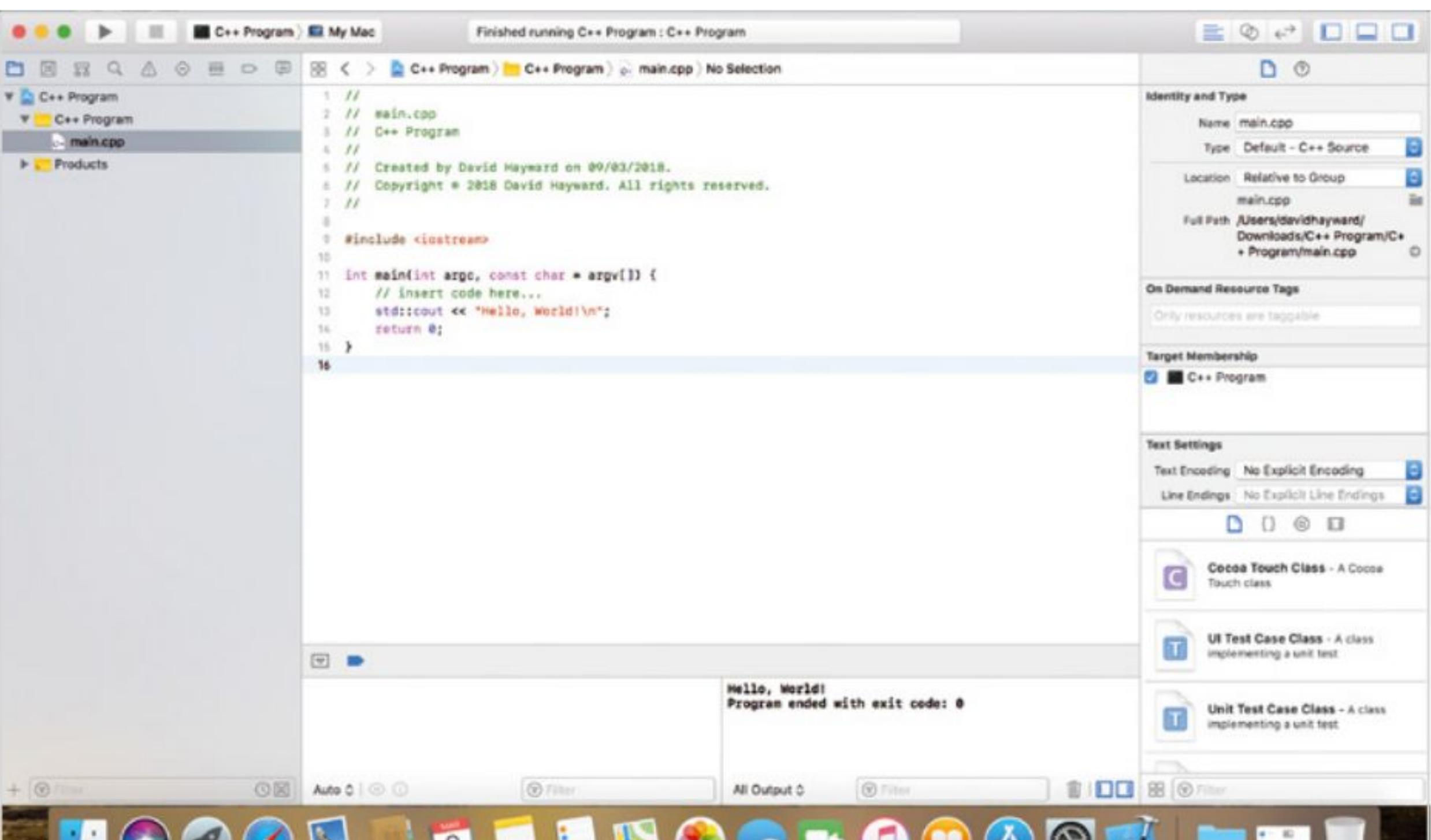
STEP 8 The next step asks where to create a Git Repository for all your future code. Choose a location on your Mac, or a network location, and click the Create button. When you've done all that, you can start to code. The left-hand pane details the files used in the C++ program you're coding. Click on the main.cpp file in the list.



STEP 9 You can see that Xcode has automatically completed a basic Hello World program for you. The differences here are that the int main () function now contains multiple functions and the layout is slightly different. This is just Xcode utilising the content that's available to your Mac.



STEP 10 When you want to run the code, click on Product > Run. You may be asked to enable Developer Mode on the Mac; this is to authorise Xcode to perform functions without needing your password every session. When the program executes, the output is displayed at the bottom of the Xcode window.



How to Set Up C++ in Linux

Linux is a great C++ coding environment. Most Linux distros already have the essential components preinstalled, such as a compiler. The text editors are also excellent for entering code into, including colour coding. There's also tons of extra software available to help you out.

LINUX++

For this example, we're using a fresh installation of Linux Mint. However, these also work if you're using Ubuntu or any other Debian-based Linux OS.

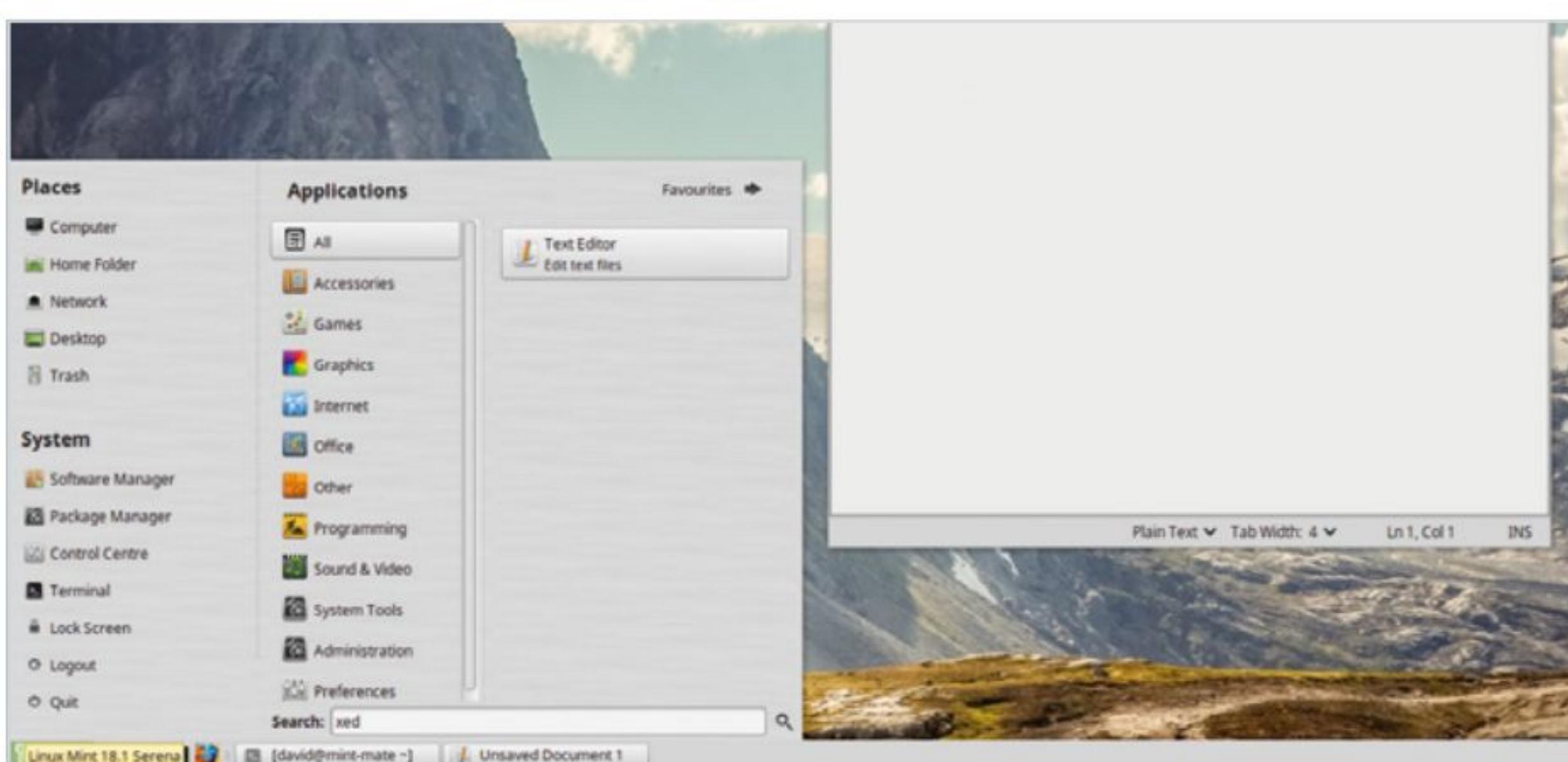
STEP 1 The first step with ensuring Linux is ready for your C++ code is check the system and software are up to date. Open a Terminal and enter: `sudo apt-get update & sudo apt-get upgrade`. Press Return and enter your password. These commands updates the entire system and any installed software.

```
david@mint-mate ~ $ sudo apt-get update &
[sudo] password for david: [REDACTED]
```

STEP 2 Most Linux distros come preinstalled with all the necessary components to start coding in C++. However, it's always worth checking to see if everything is present, so still within the Terminal, enter: `sudo apt-get install build-essential` and press Return. If you have the right components, nothing is installed but if you're missing some then they are installed by the command.

```
david@mint-mate ~ $ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
build-essential set to manually installed.
0 to upgrade, 0 to newly install, 0 to remove and 0 not to upgrade.
david@mint-mate ~ $
```

STEP 3 Amazingly, that's it. Everything is all ready for you to start coding. Here's how to get your first C++ program up and running. In Linux Mint the main text editor is Xed can be launched by clicking on the Menu and typing **Xed** into the search bar. Click on the Text Editor button in the right-hand pane to open Xed.

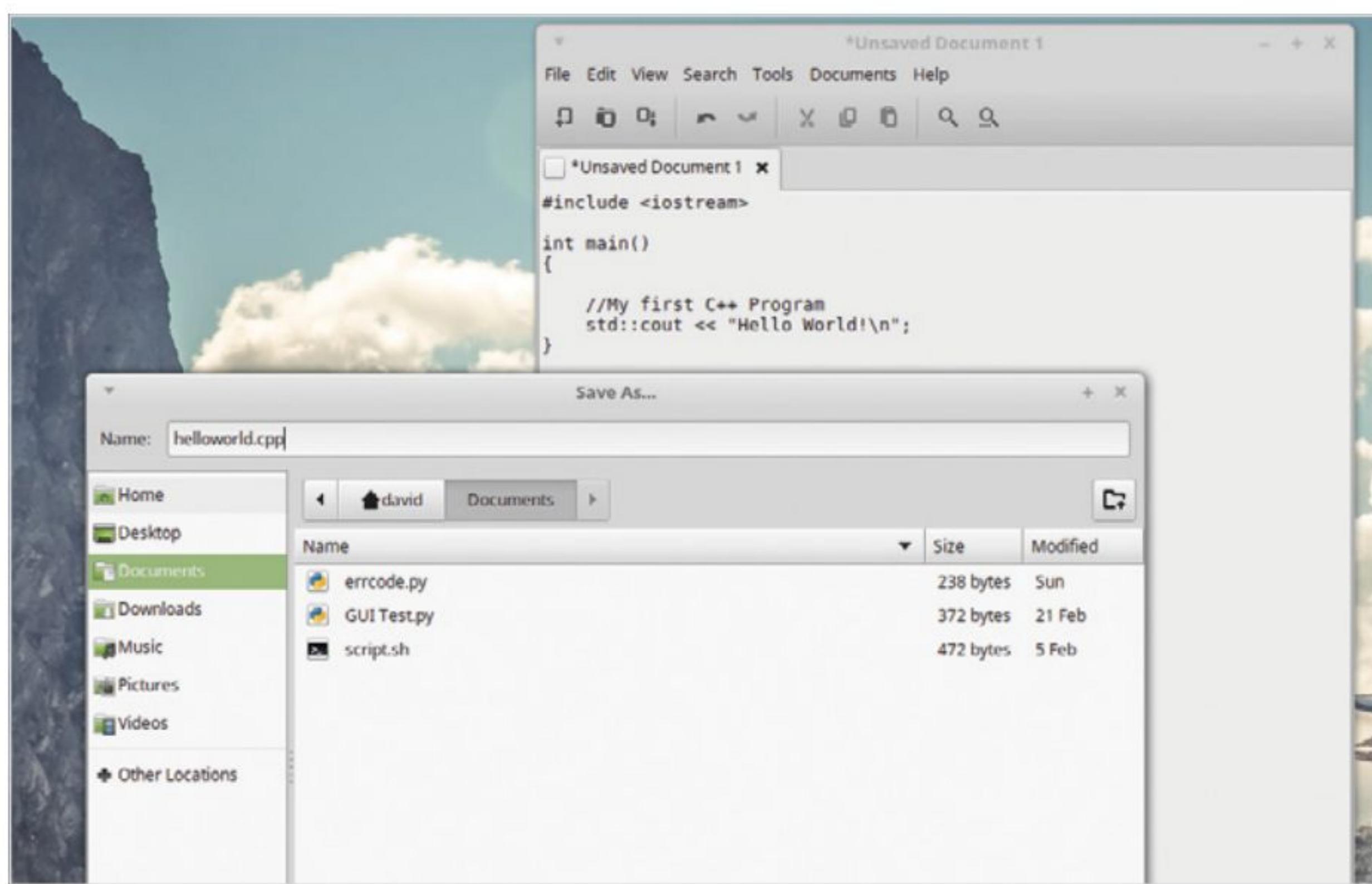


STEP 4 In Xed, or any other text editor you may be using, enter the lines of code that make up your C++ Hello World program. To remind you, its:

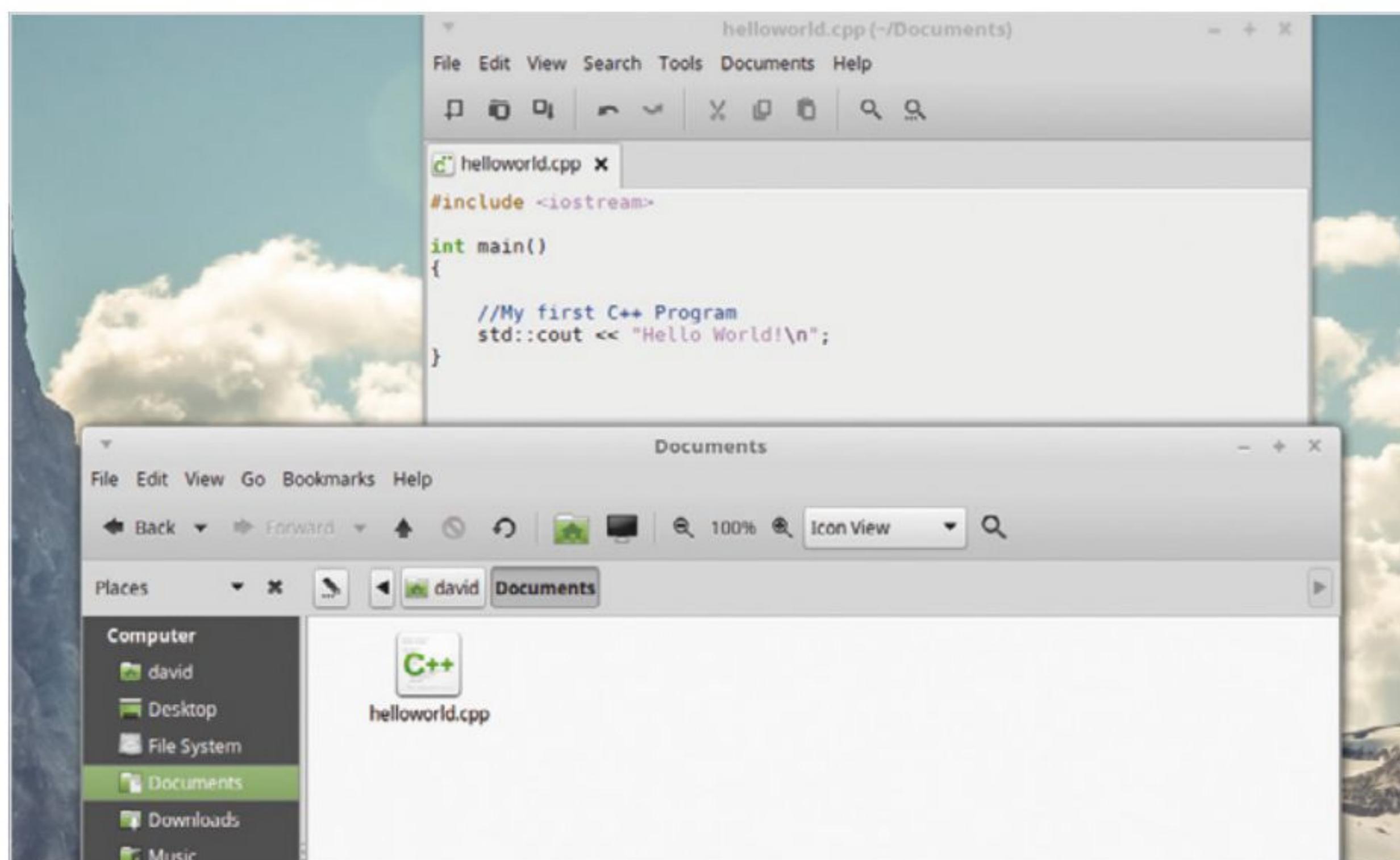
```
#include <iostream>
int main()
{
    //My first C++ program
    std::cout << "Hello World!\n";
}
```

STEP 5

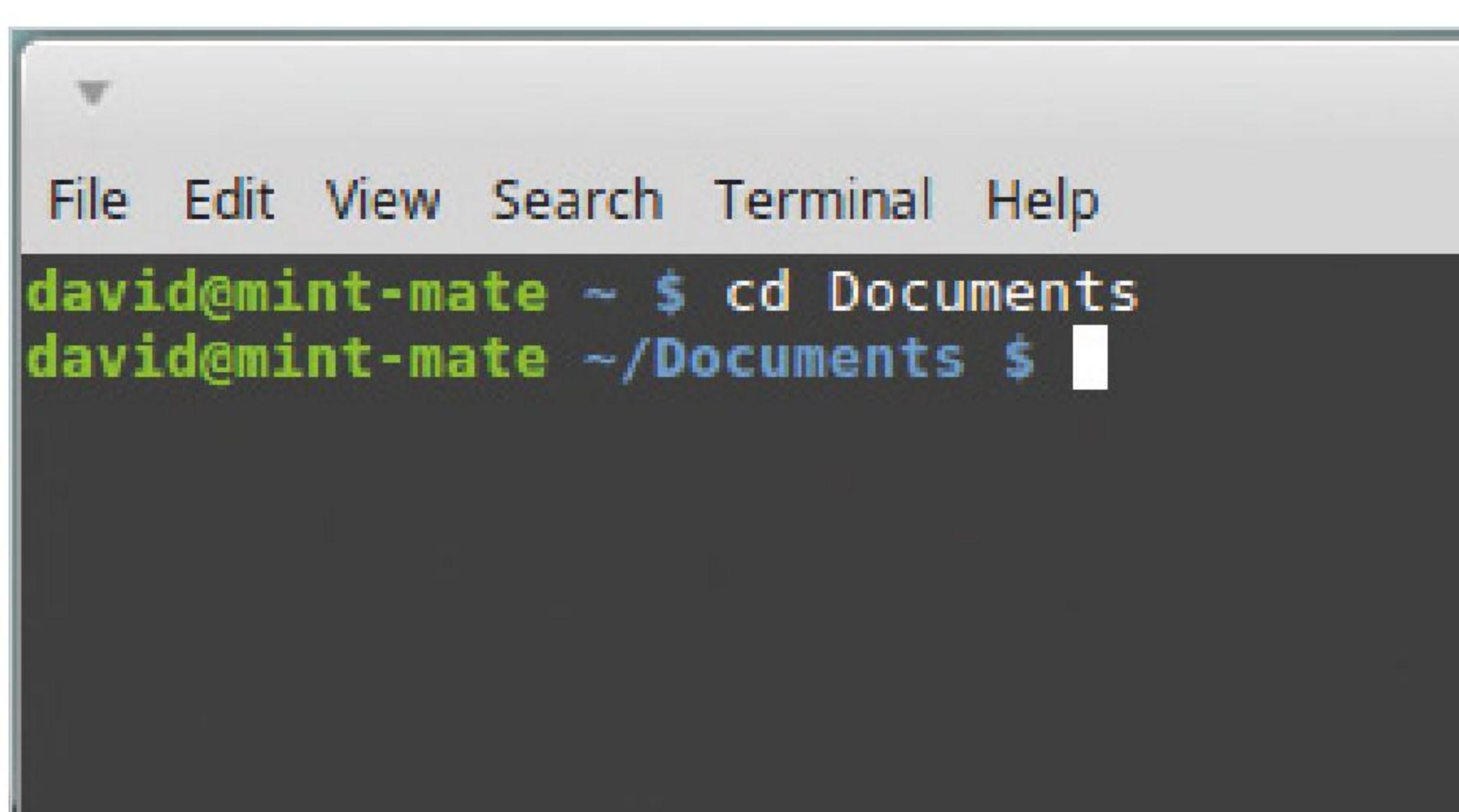
When you've entered your code, click File > Save As and choose a folder where you want to save your program. Name the file as helloworld.cpp, or any other name just as long as it has .cpp as the extension. Click Save to continue.

**STEP 6**

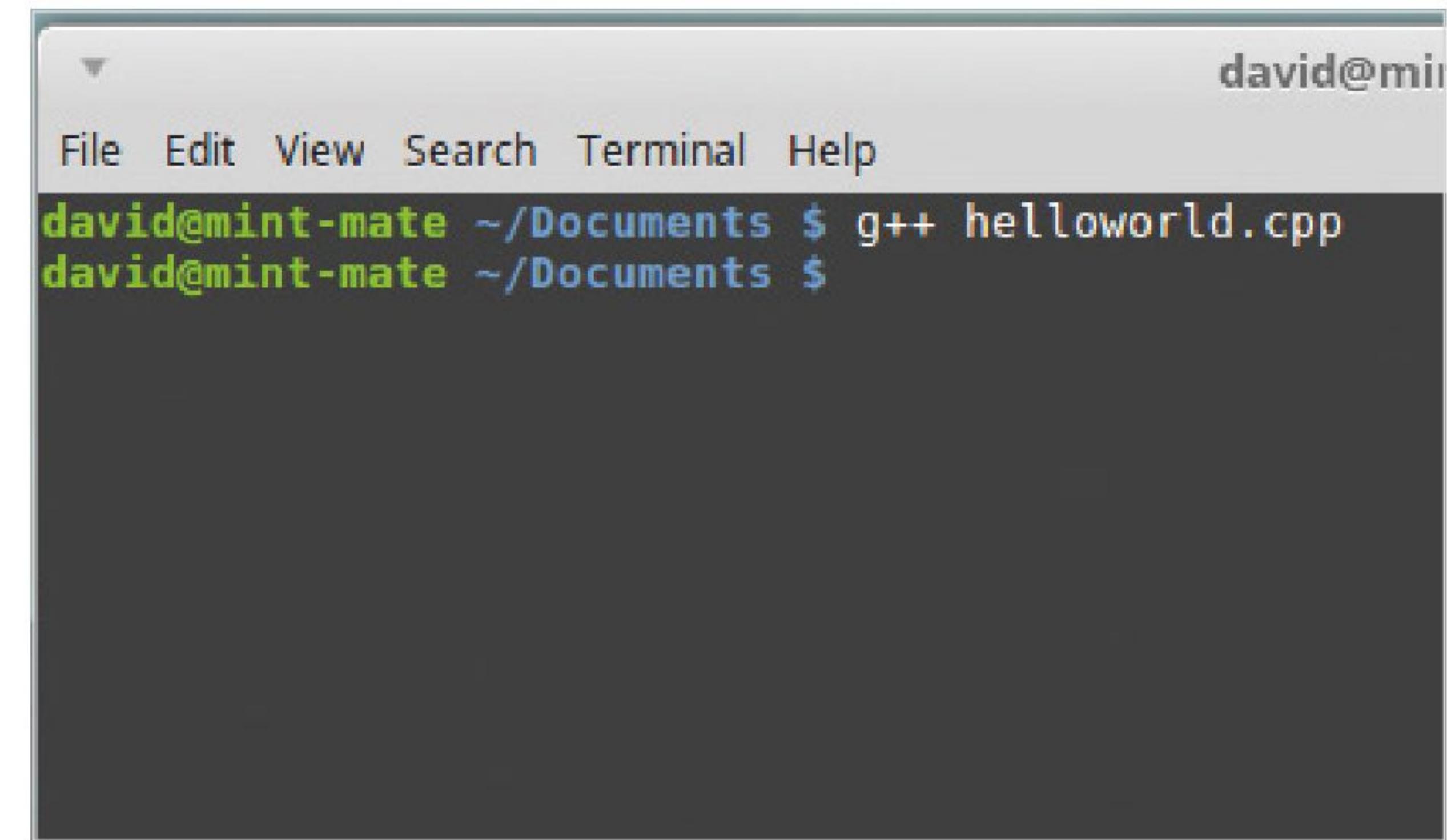
The first thing you can see is that Xed has automatically recognised this as a C++ file, since the file extension is now set to .cpp. The colour coding is present in the code and if you open up the file manager you can also see that the file's icon has C++ stamped on it.

**STEP 7**

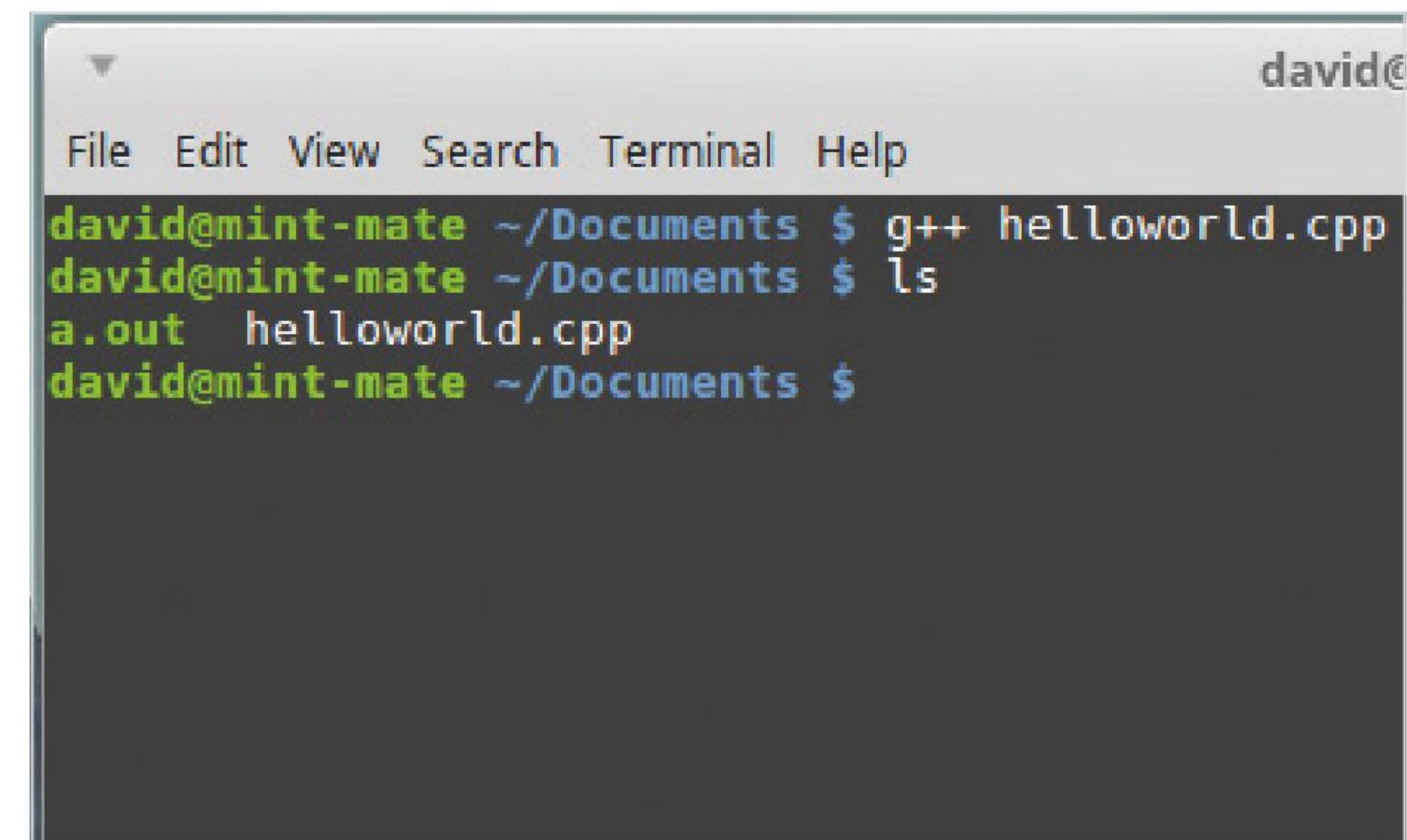
With your code now saved, drop into the Terminal again. You need to navigate to the location of the C++ file you've just saved. Our example is in the Documents folder, so we can navigate to it by entering: `cd Documents`. Remember, the Linux Terminal is case sensitive, so any capitals must be entered correctly.

**STEP 8**

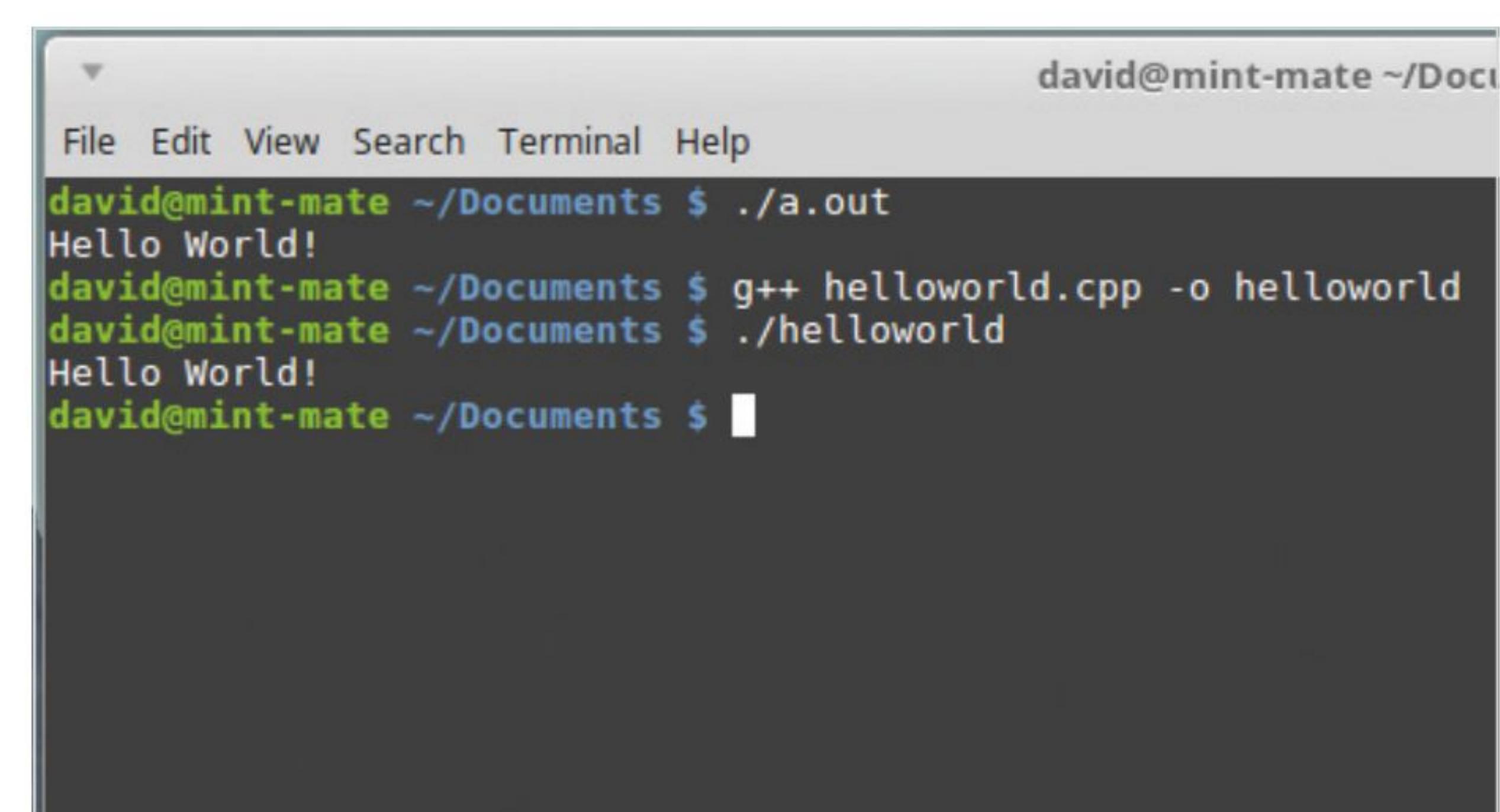
Before you can execute the C++ file you need to compile it. In Linux it's common to use g++, an open source C++ compiler and as you're now in the same folder as the C++ file, go to the Terminal, enter: `g++ helloworld.cpp` and press return.

**STEP 9**

There will be a brief pause as the code is compiled by g++ and providing there are no mistakes or errors in the code you are returned to the command prompt. The compiling of the code has created a new file. If you enter `ls` into the Terminal you can see that alongside your C++ file is a.out.

**STEP 10**

The a.out file is the compiled C++ code. To run the code enter: `./a.out` and press Return. The words 'Hello World!' appears on the screen. However, a.out isn't very friendly. To name it something else post-compiling, you can recompile with: `g++ helloworld.cpp -o helloworld`. This creates an output file called helloworld which can be run with: `./helloworld`.



Other C++ IDEs to Install

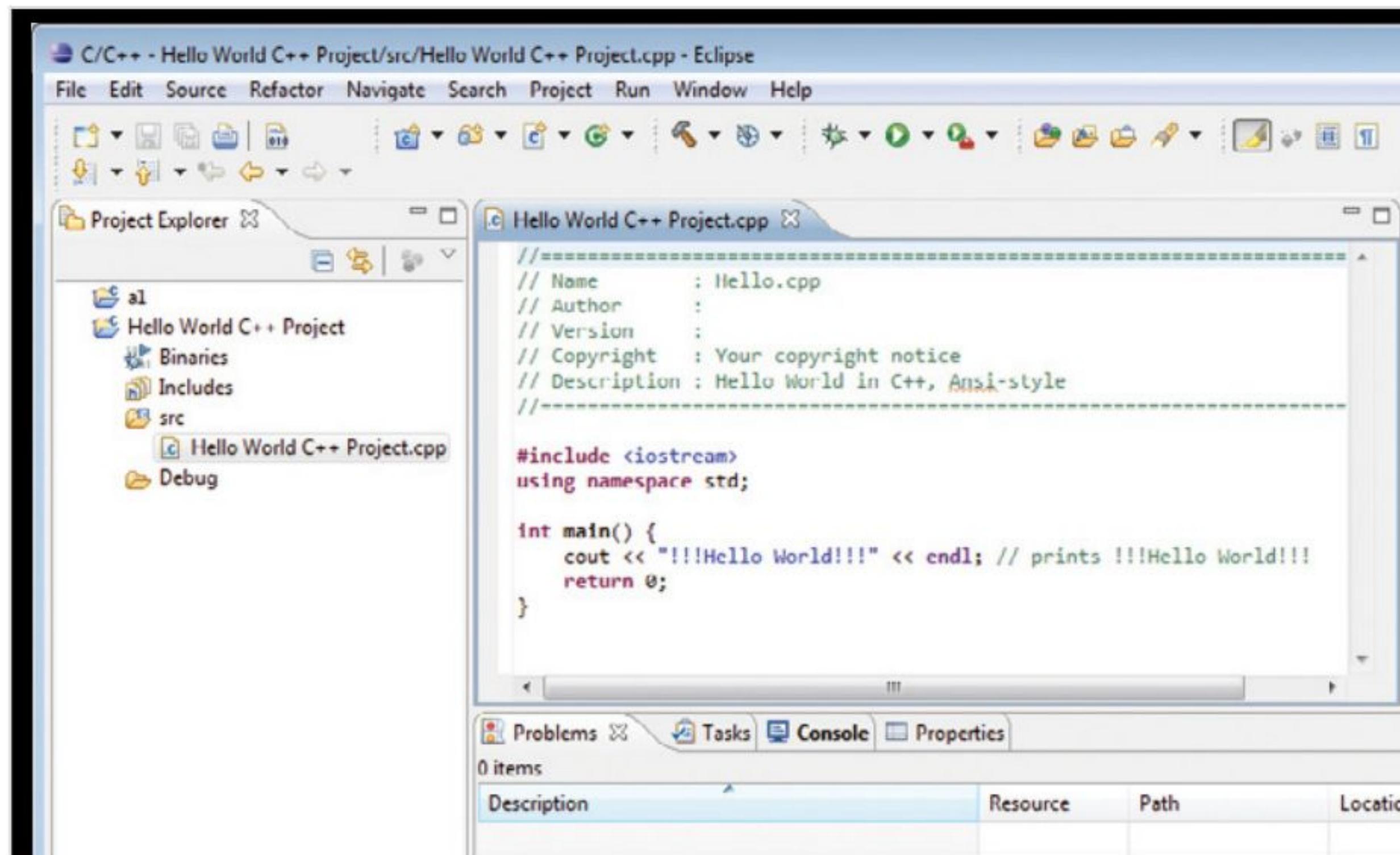
If you want to try a different approach to working with your C++ code, then there are plenty of options available to you. Windows is the most prolific platform for C++ IDEs but there are plenty for Mac and Linux users too.

DEVELOPING C++

Here are ten great C++ IDEs that are worth looking into. You can install one or all of them if you like, but find the one that works best for you.

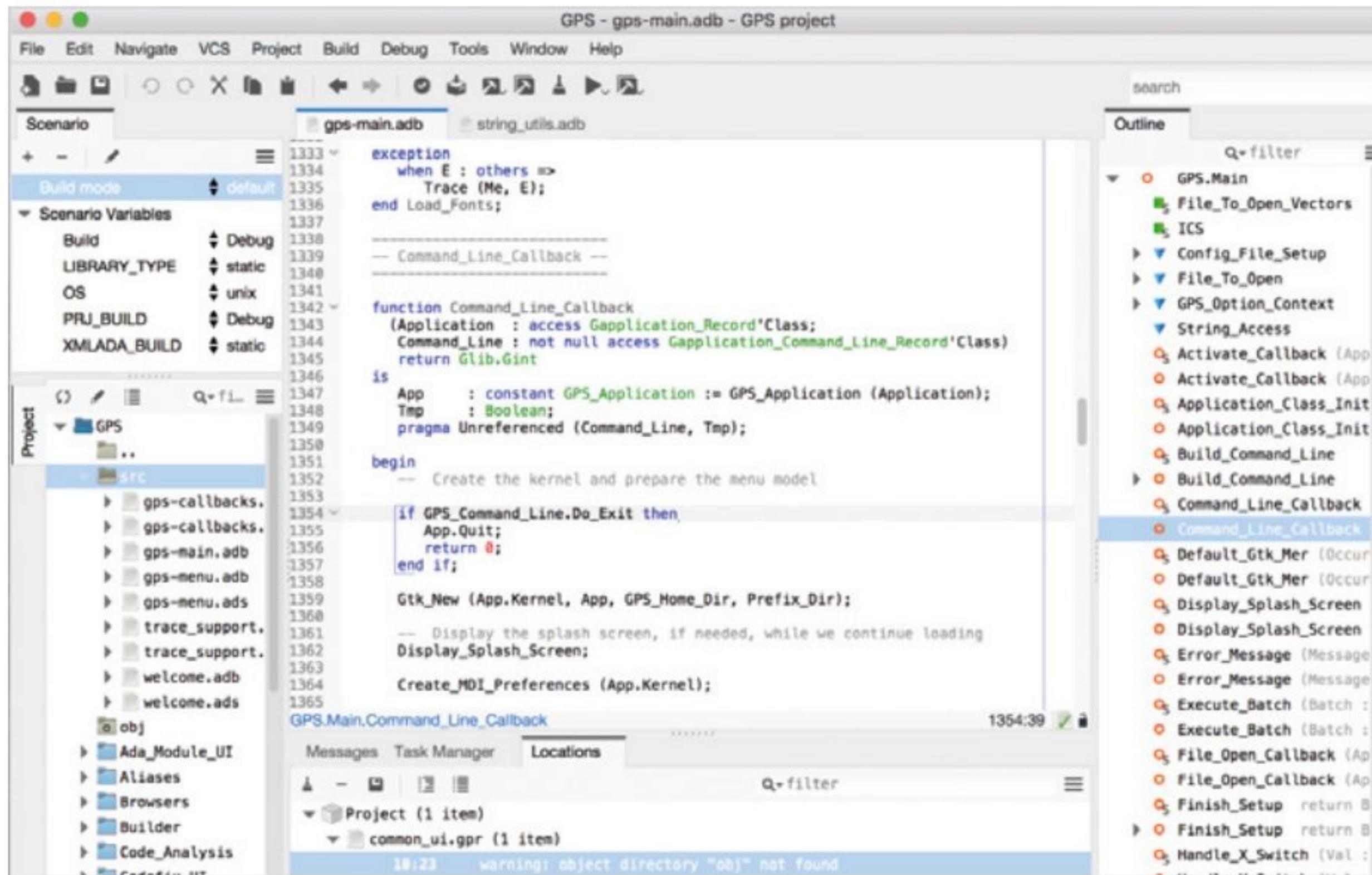
ECLIPSE

Eclipse is a hugely popular C++ IDE that offers the programmer a wealth of features. It has a great, clean interface, is easy to use and available for Windows, Linux and Mac. Head over to www.eclipse.org/downloads/ to download the latest version. If you're stuck, click the Need Help link for more information.



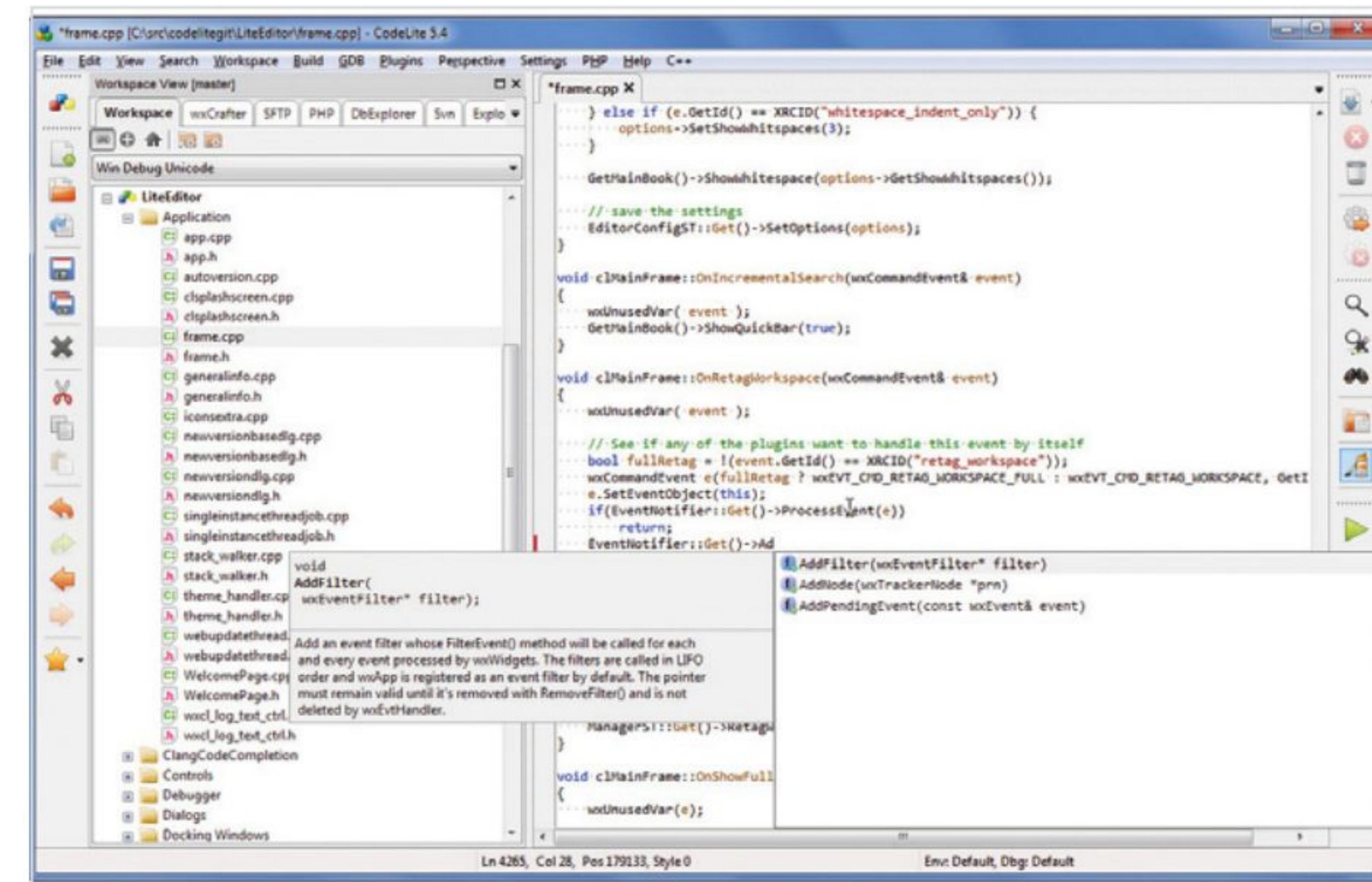
GNAT

The GNAT Programming Studio (GPS) is a powerful and intuitive IDE that supports testing, debugging and code analysis. The Community Edition is free, whereas the Pro version costs; however, the Community Edition is available for Windows, Mac, Linux and even the Raspberry Pi. You can find it at www.adacore.com/download.



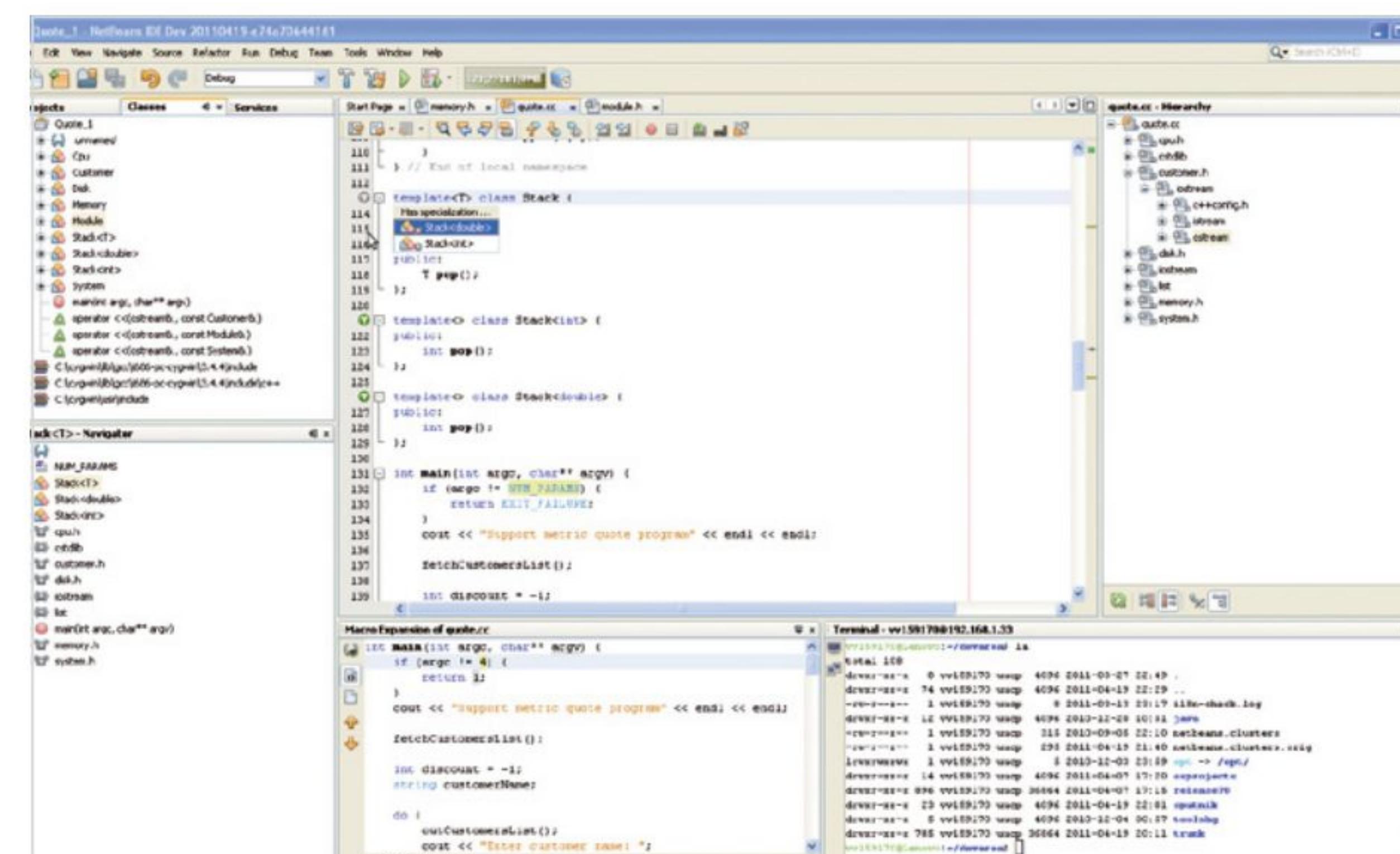
CODELITE

CodeLite is a free and open source IDE that's regularly updated and available for Windows, Linux and macOS. It's lightweight, uncomplicated and extremely powerful. You can find out more information as well as how to download and install it at www.codelite.org/.



NETBEANS

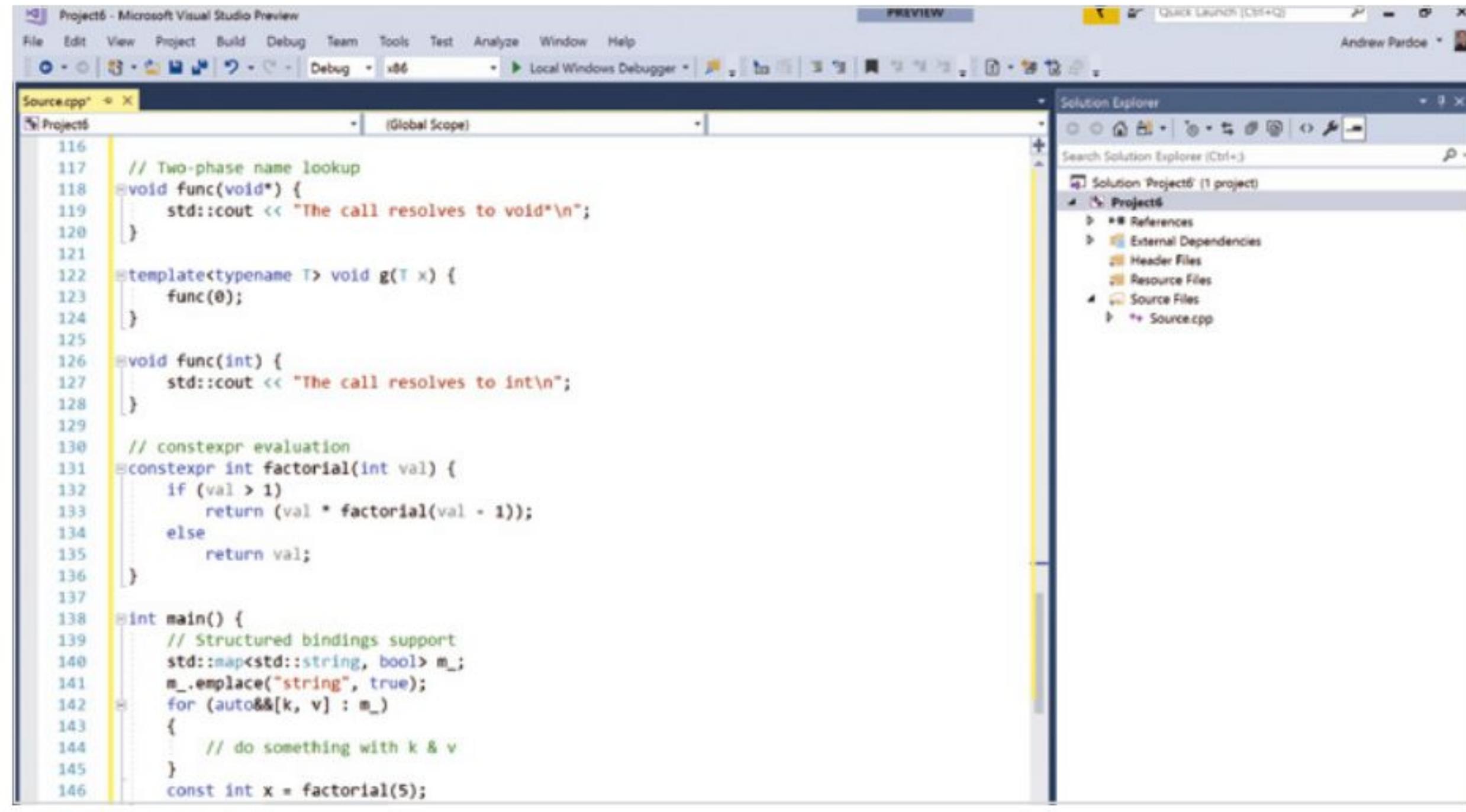
Another popular choice is NetBeans. This is another excellent IDE that's packed with features and a pleasure to use. NetBeans IDE includes project based templates for C++ that give you the ability to build applications with dynamic and static libraries. Find out more at www.netbeans.org/features/cpp/index.html.





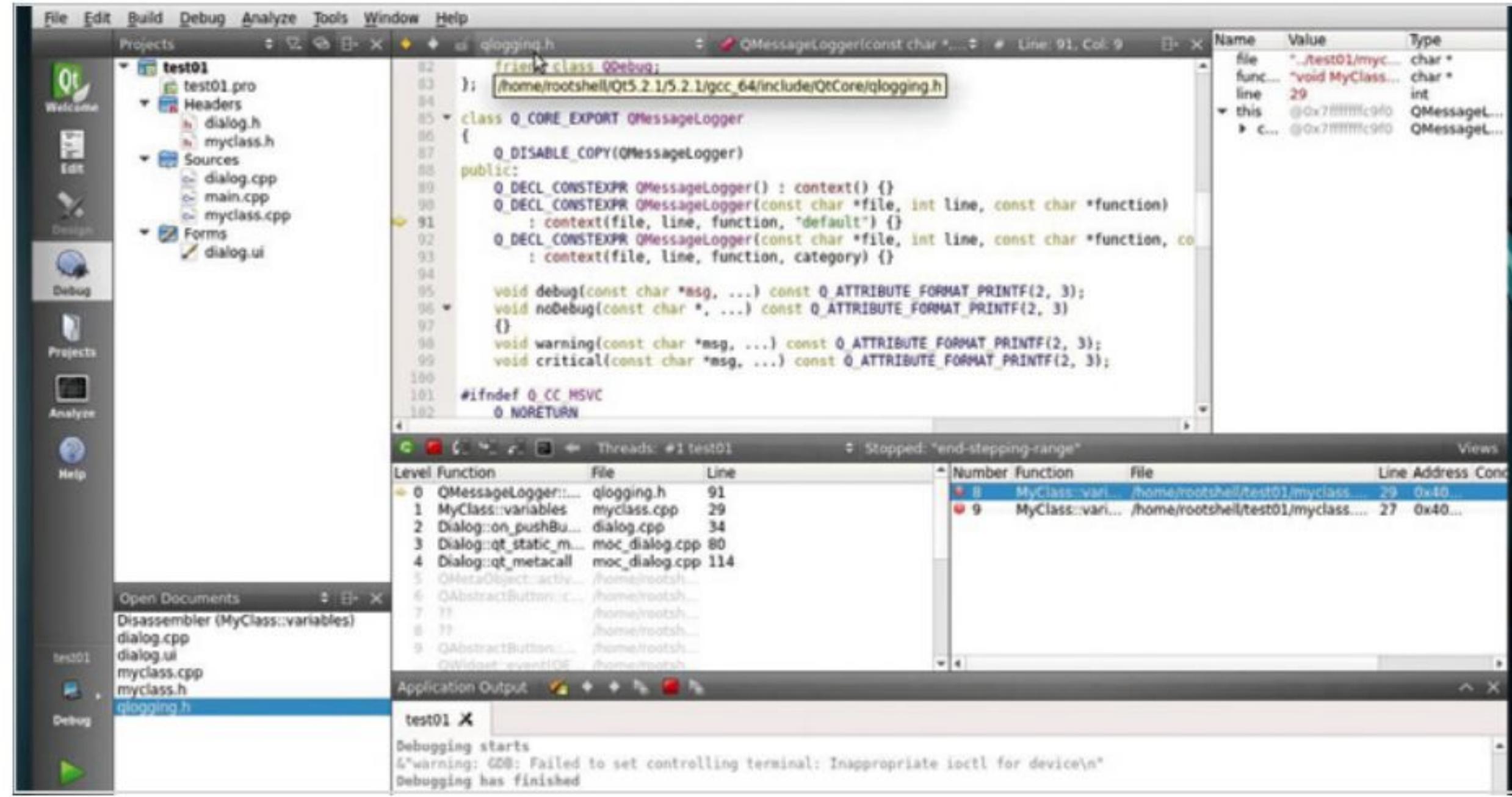
VISUAL STUDIO

Microsoft's Visual Studio is a mammoth C++ IDE that allows you to create applications for Windows, Android, iOS and the web. The Community version is free to download and install but the other versions allow a free trial period. Go to www.visualstudio.com/ to see what it can do for you.



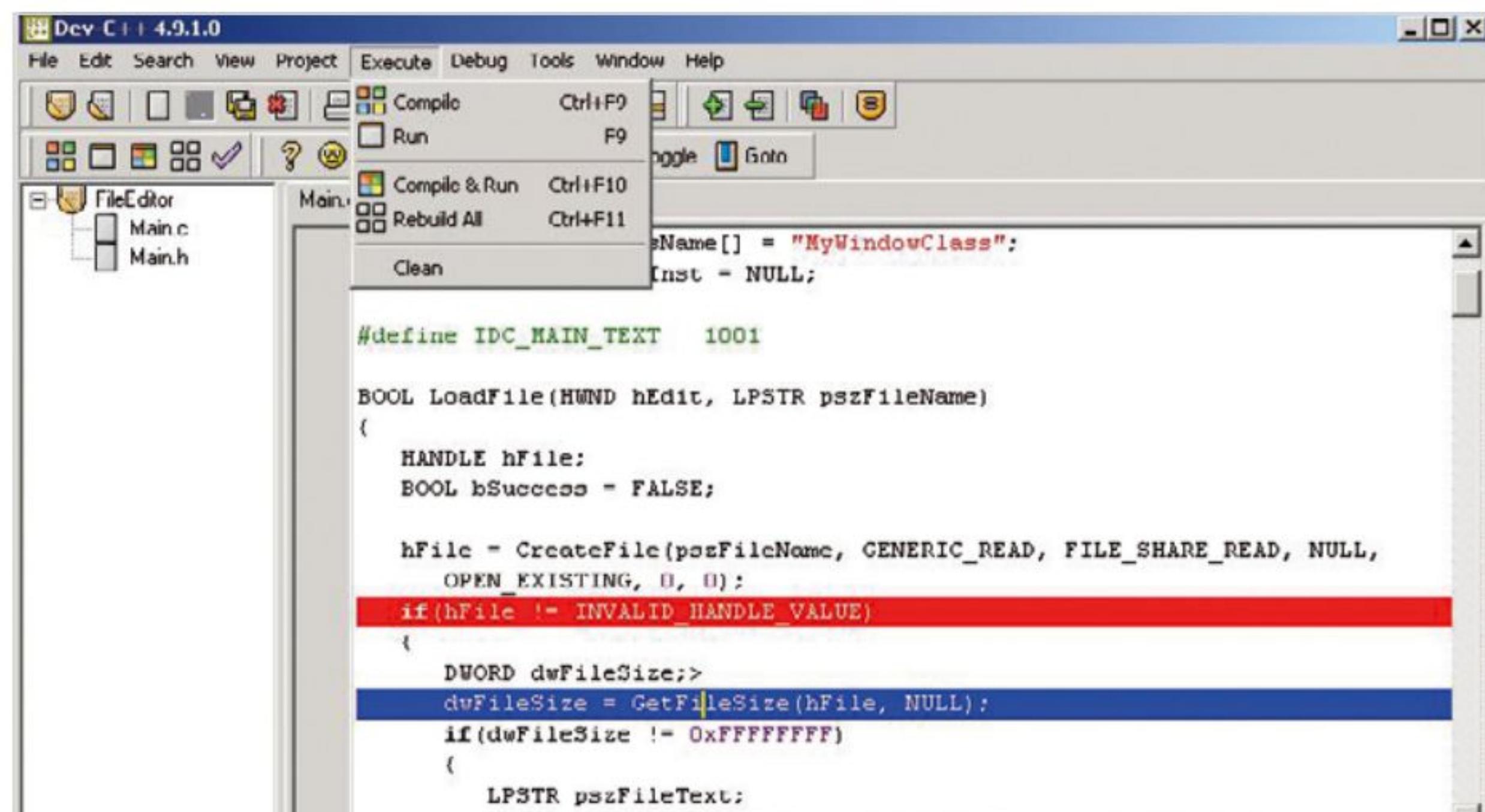
QT CREATOR

This cross-platform IDE is designed to create C++ applications for desktop and mobile environments. It comes with a code editor and integrated tools for testing and debugging, as well as deploying to your chosen platform. It's not free but there is a trial period on offer before requiring purchasing: www.qt.io/qt-features-libraries-apis-tools-and-ide/.



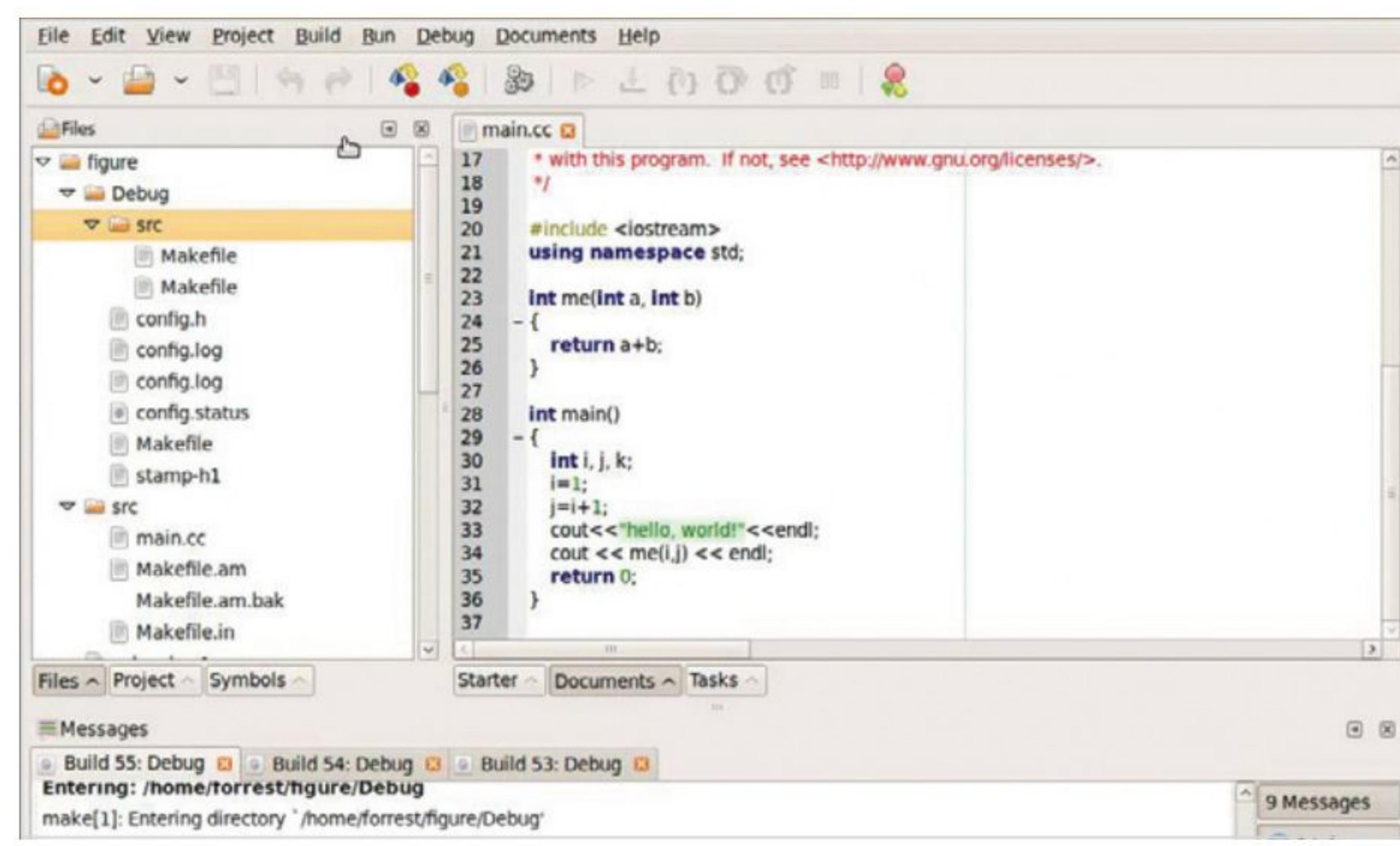
DEV C++

Bloodshed Dev C++, despite its colourful name, is an older IDE that is for Windows systems only. However, many users praise its clean interface and uncomplicated way of coding and compiling. Although there's not been much updating for some time, it's certainly one to consider if you want something different: www.bloodshed.net/devcpp.html.



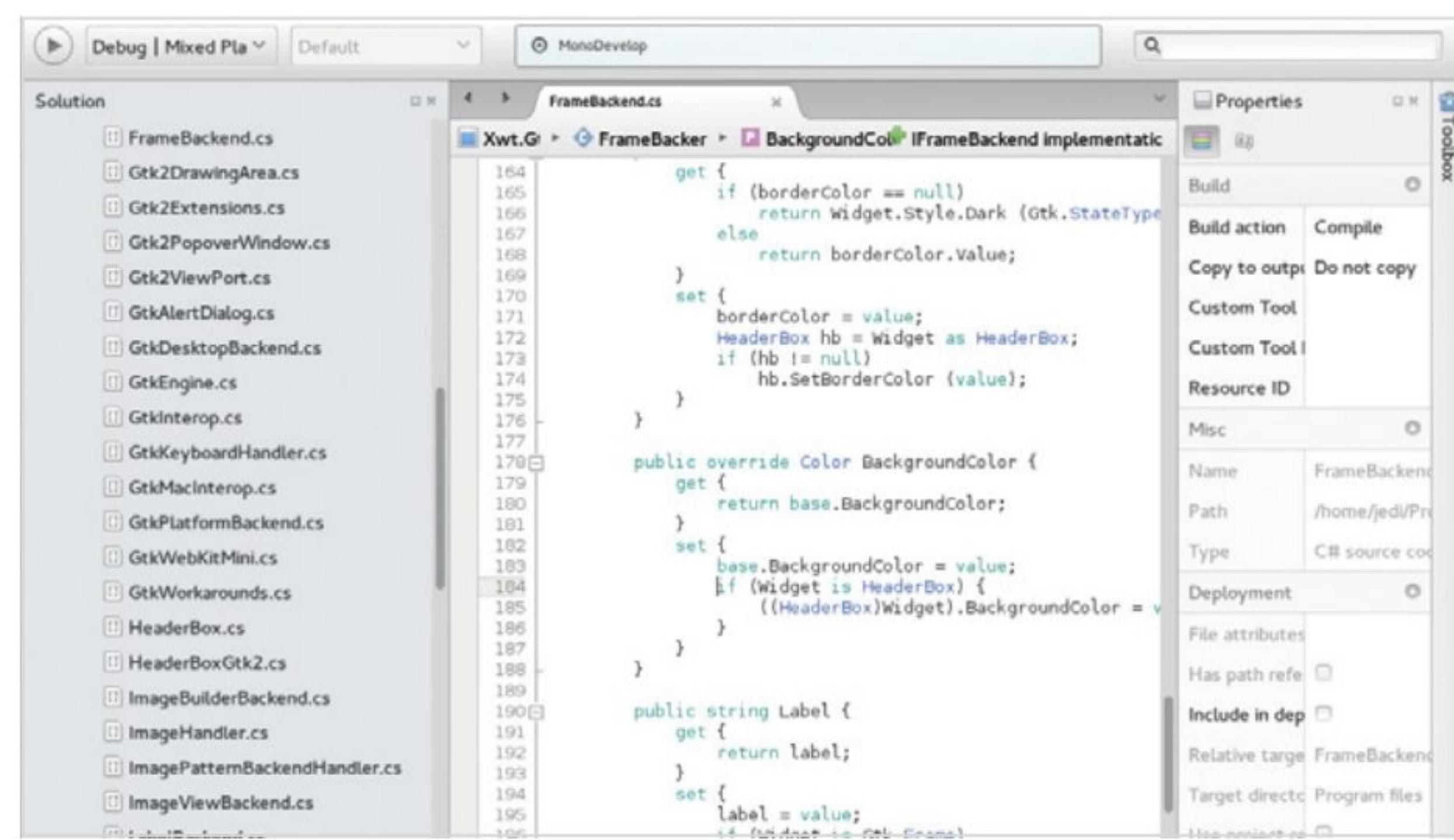
ANJUTA

The Anjuta DevStudio is a Linux-only IDE that features some of the more advanced features you would normally find in a paid software development studio. There's a GUI designer, source editor, app wizard, interactive debugger and much more. Go to www.anjuta.org/ for more information.



MONODEVELOP

This excellent IDE allows developers to write C++ code for desktop and web applications across all the major platforms. There's an advanced text editor, integrated debugger and a configurable workbench to help you create your code. It's available for Windows, Mac and Linux and is free to download and use: www.monodevelop.com/.



U++

Ultimate++ is a cross-platform C++ IDE that boasts a rapid development of code through the smart and aggressive use of C++. For the novice, it's a beast of an IDE but behind its complexity is a beauty that would make a developer's knees go wobbly. Find out more at www.ultimatepp.org/index.html.

