



**European
University Cyprus**

LAUREATE INTERNATIONAL UNIVERSITIES

School of Sciences – Department of Computer Science and Engineering

**Programming Principles 1 (CSC131) &
Programming Principles 2 (CSC132)
Quick software guide**

Installing and using DevC++ on Windows

| | |
|--|----------|
| WHAT SOFTWARE IS NEEDED | 2 |
| WHERE DO I FIND THE SOFTWARE? | 2 |
| HOW DO I INSTALL THE SOFTWARE? | 2 |
| DO I NEED TO CONFIGURE ANYTHING? | 2 |
| HOW DO I USE DEVC++ FOR MY PROGRAMS | 3 |
| PROGRAMMING TIPS | 6 |

© Andreas Grondoudis for DCSE

The following assumes that you already have a machine with a current version of MS Windows¹ installed.

What software is needed

One piece of software is required. It is a free to use C and C ++ integrated development environment (IDE) called DevC++. At the time of this writing, the latest version is 5.8.1.

Being an IDE means that once you install this piece of software you will have both the language tools required as well as an editor to write your programs in.

Frequently Asked Questions

Q: Do I have a choice? Can I use something else?

A: Of course. There are 10s of free editor and tools available. Search the internet, find one you like, install it and use it. However, it is not possible for me to assist you in any of these steps. You can even install any distribution of Linux and use the <g++> compiler.

Where do I find the software?

Visit <http://orwelldevcpp.blogspot.com/> and use the 'Blog Archive' to find a post about the latest release. Download links will be available on the post. You can download the software from there.

Frequently Asked Questions

Q: What size is the download?

A: At last writing: about 47MB

How do I install the software?

DevC++ comes with a standard windows installer. Just run the application, follow the sequence of steps and in a short amount of time (and a few clicks) you should be finished. If you use the default settings a shortcut will be placed on your desktop and you can start the application from there.

Do I need to configure anything?

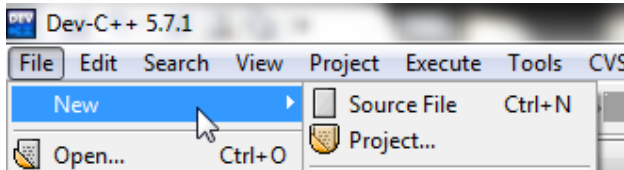
No, DevC++ being an IDE means that it knows where everything is and it configured itself during installation time. In effect, once the installation is finished you are good to go... just write your programs and run them.

¹ This guide has been created using Windows 7

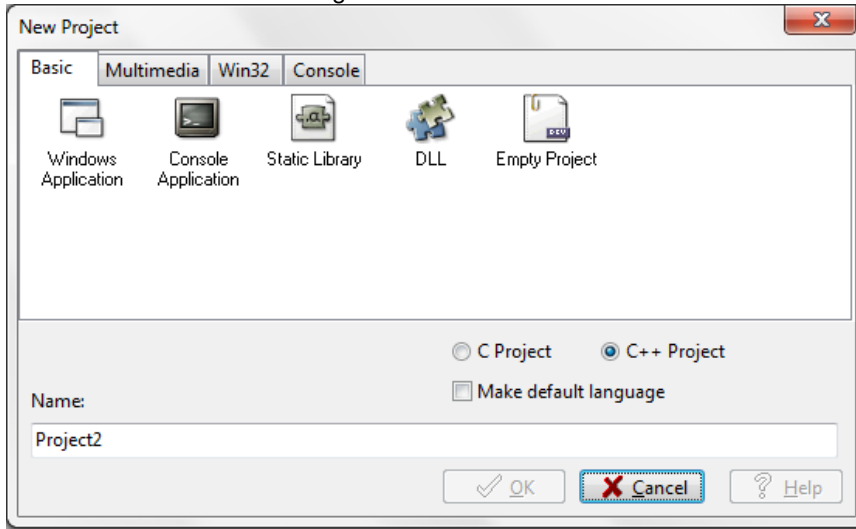
How do I use DevC++ for my programs

DevC++ operates on the basis of a project. You create a project, select the type of project and then you write the source files for the project. You then compile the project and if successful run the project...i.e. your files.

Start the application and use the <File> menu, select <New> and click on <Project>. The capture below demonstrates:



The <New Project> dialog will appear and here you must select the type of project you want. The capture below demonstrates the dialog



For CSC131 (using C)

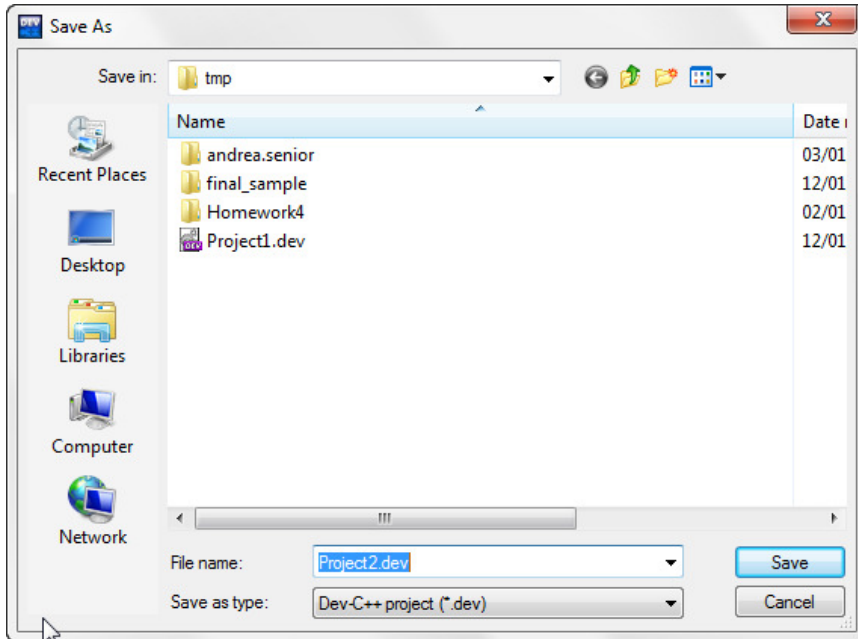
- Select <Console Application>
- Select <C project>
- Enter a project <Name:>

For CSC132 (using C++)

- Select <Empty Project>
- Select <C++ Project>
- Enter a project <Name:>

Once you select the type, the <OK> button will be enabled. Click it. DevC++ will then present you with a dialog to select the location of where you want the project created (as well as the name of the project file).

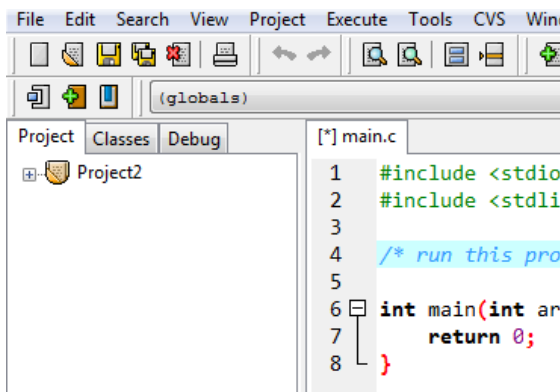
The capture below demonstrates:



Select the location you want and click <Open>. DevC++ will create the project for you and open it up so you can start writing your source files.

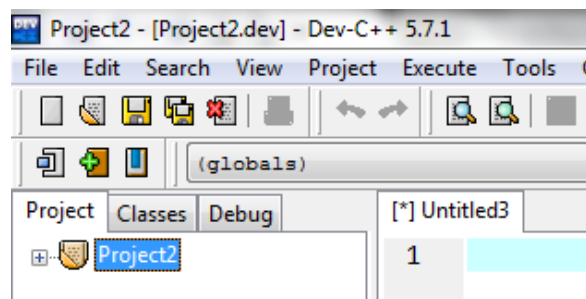
For CSC131 (using C)

The project is created, a <main.c> file is created and opened for you to edit. The capture below shows the project folder and the file opened up for editing.

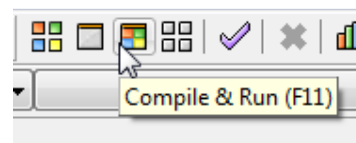
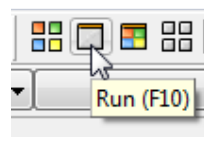


For CSC132 (using C++)

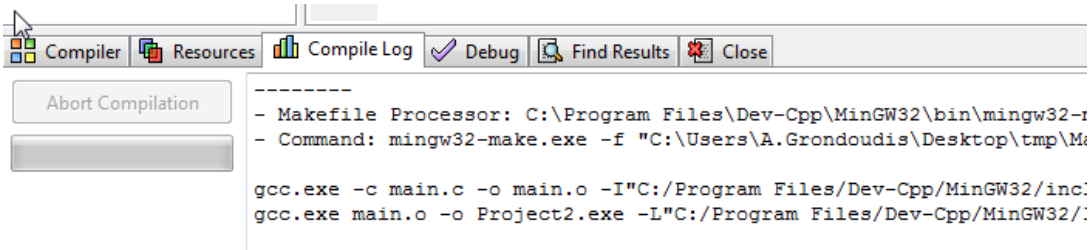
The project is created, an empty file is created and opened for you. The capture below shows the project folder and the untitles3 file opened up for editing.



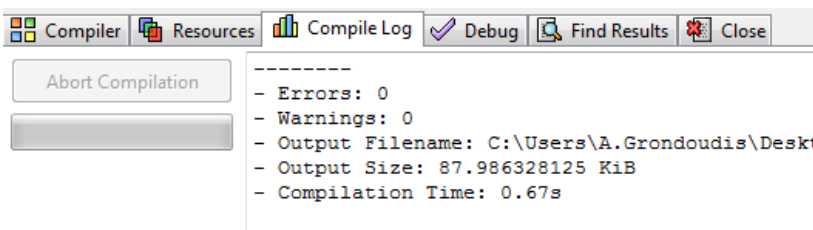
Once you write the file you need to compile and then (if the compilation is successful) you need to run the program to see if it produces the required result. The toolbar at the top of the window includes 3 buttons that enable you to command the IDE. The captures below demonstrate:



When you click compile the bottom of the IDE shows the <compile log> and you can actually see the process of compiling executing.



If everything is ok, the compile log will finish its scrolling and display something like the capture below. Then you can run the program to get the output.



However, if the compiler finds one or more mistakes the bottom of the IDE will switch to the <Compiler> tab and display a list of the errors At the same time, the editor will switch to the file and highlight the line in question so that you (the programmer) can correct the mistake. The captures below demonstrate the error listing (top) and the highlighted code (bottom).

The screenshot shows the 'Compiler' window in an IDE. The window has a title bar with 'Compiler (3)', 'Resources', 'Compile Log', 'Debug', 'Find Results', and 'Close' buttons. Below the title bar is a table with the following data:

| Line | Col | File | Message |
|------|-----|--|--------------------------------------|
| | | C:\Users\A.Grondoudis\Desktop\tmp\main.c | In function 'main': |
| 10 | 2 | C:\Users\A.Grondoudis\Desktop\tmp\main.c | [Error] expected ';' before 'return' |
| 28 | | C:\Users\A.Grondoudis\Desktop\tmp\Makefile.win | recipe for target 'main.o' failed |
| | | | |
| | | | |

The screenshot shows the code editor in an IDE. The code is as follows:

```
7   int i=0;  
8   for (i=0 ; i<34 ; i++)  
9     printf("Hello World")  
10  return 0;  
11  }
```

The line `return 0;` is highlighted in red, indicating an error.

Programming tips

1. You have to be patient and persistent. Programs can take a while to complete. Don't give up
2. The compiler has nothing against you, as do the keyboard, monitor and case. They are just doing their job; don't take your frustration out on them.
3. The compiler is very rarely wrong. If it is reporting a mistake then it means that you do have a mistake somewhere
4. When you do get 10 or even 100 scrolls worth of mistakes do NOT despair. Usually mistakes at the beginning of the program can cause many mistakes later on. What you should do is scroll-back at the top of the compiler output window and try to correct the mistakes starting from the top.
5. Quite often mistakes are reported on lines that when you see them no mistake exists. This, 99% of the times, means that there is mistake somewhere above in the program. If the code is in a loop somewhere; look at the loop first and then look and the method that the loop was in and so on. Try and double-check related code (close and sometimes far) to the mistake reporting line.
6. When you do copy and paste code that you don't want to retype you have to take care to make any minor required changes. Quite often, especially when we have to paste the code in more than one place what you do is: paste everywhere first and then go back and adjust the code. The problem here is that you might have actually copied the code in, say, 10 places and went to make the adjustments in only 9 of them. Then the one place that you didn't change will: 1) Not report an error and 2) Run OK, but quite possibly not as you want it. So, what you need to do is, paste the code and make the changes there and then (so that you don't forget later on). The clipboard holds the text so you can then carry on to the next paste and change.
7. When you are starting a method or a loop or anything else that might require the use of start and end braces what most people do is start with the open brace and carry on to type the code inside the brace. Then (at the end) type the closing brace. This is fine but, if you have to type many lines inside the brace, you tend to forget to put the closing brace at the end. This leads to the compiler reporting (numerous sometimes) '}' expected'. A very handy tip to use is: Write both the opening and closing brace at the same time and then take your cursor back in the middle of the two to complete the code. This way, anything you write in there (no matter how many lines it is) will be inside the braces.