

## Working with Projects (Dev-C++)

Dev-C++ can be downloaded free of charge from:

<http://www.bloodshed.net/dev/devcpp.html>. You want to download: **Dev-C++ 5.0 beta 9.2 (4.9.9.2) (9.0 MB) with Mingw/GCC 3.4.2** from SourceForge. (This is the first download link.)

Now you want to configure Dev-C++ appropriately for this course by doing the following: Start Dev-C++ and go to the “Tools” menu and select “Compiler Options”. ~~First, under the “Compiler” tab, in the box under “Add the following commands when calling compiler” add: “-Wno-deprecated” (without the quotes and with no spaces) and ensure that there is a tick mark next to “Add the following commands when calling compiler”.~~ (Note that the `-Wno-deprecated` should no longer be needed.) Second, under the “Settings” tab, click on “Linker” and ensure that after “Generate debugging information” it says “Yes” (and not “No”).

Here’s how to create a project in Dev-C++. Under the “File” menu, select “New” and then “Project”. Under the “Basic” tab, select “Console Application”. Ensure that “C++ Project” is selected in the “Project options”, and choose a name for your project (overwriting the supplied default of “Project1”), let’s say “XXXX”. Your executable will be given the project name, i.e. “XXXX.dev”. You will now be prompted for the directory where you want to store the project file. If you already have some files written for this project, then choose the directory where they are stored. Otherwise pick a new directory in an appropriate place. Once you have selected the directory, click “Save”.

You now want to add your .cpp files to your project. Under the “Project” menu select “Add to project” (or use the shortcut) and select each .cpp file that you want in your project. It will make your life simple if you have all these .cpp files in the same directory as the project file. Dev-C++ may have created a main.cpp file that you do not need. If so, remove it using “Remove from project” in the “Project” menu (or the shortcut). You do not need to save this file. Note that only .cpp files have to be added to your project. Header (.h) files are implicitly included through being referenced in your .cpp files. Just ensure that your header files are in the same directory as your project file and .cpp files.

Under the “Project” tab, you should be able to see a list of the .cpp files included in your project. If not, ensure that there is a “-” next to the name of your project. When you click on a .cpp file, that will open it and you will be able to view it on the right hand side of the Dev-C++ window. You can switch back and forth between open files by clicking the tabs above the display window.

Once all the necessary files have been created and added to your project, you can use the “Execute” menu (or shortcuts) to “Compile” (all files), “Compile current file”, “Run” (most recently linked version), and “Compile and run” (compile all files, link and run) as required.

To return to a previously created project, you can browse to the project directory and double-click on “XXXX.dev” (where “XXXX” is the name of your project). This should bring up Dev-C++ and leave you exactly where you were when you were last working on the project. Or, alternatively, you can select use the “File” menu + “Open Project or File” while already in Dev-C++ (and then select “XXXX.dev”). The net effect will be the same.

While browsing the files in your project folder, note that Dev-C++ gives the object files an “.o” file type, rather than “.obj” as in most other C++ development environments.