

How to install Code::Blocks

Alexandra Stefan

Code::Blocks and C compiler

- C compiler
 - In order to run C code a C compiler is needed.
 - Programmers write code in a “human readable form”. A compiler will generate a corresponding special program that the computer can run.
- IDE (Integrated Development Environment)
 - is a program that is used to edit, compile, run and debug code, BUT it still needs a C compiler in order to do that.
- Code::Blocks is an IDE. It is NOT a C compiler
 - You need both Code::Blocks itself and a C compiler.
 - I have selected Code::Blocks because it has an option to download and install at the same time both the IDE and a C compiler. It has other options that do NOT include the compiler so pay attention to which installer you download.
- Other IDEs are available (e.g. Apache NetBeans, Microsoft Visual Studio).
 - Any IDE is fine, but you need to have a C compiler and set-up the IDE to find it.

Downloading Code::Blocks

From the Code::Blocks download page: <http://www.codeblocks.org/downloads>

Click “Download the binary release”



Unix and Mac

- Mac users:
 - For C compiler: install XCode from the AppStore
 - Download and install Code::Blocks for Mac
- Unix/Linux users:
 - Install the GCC (GNU Compiler Collection) compiler – E.g. search “install gcc compiler on Unix” and follow instructions that seem clear to you.
 - Download and install Code::Blocks for your system

Windows

Download **codeblocks-20.03mingw-setup.exe** - this is the package that has both **Code::Blocks** and a **C compiler (the MinGW C compiler)**. If you already have a C compiler or prefer to install the C compiler separate, download the appropriate package (e.g. codeblocks-20.03-setup.exe).

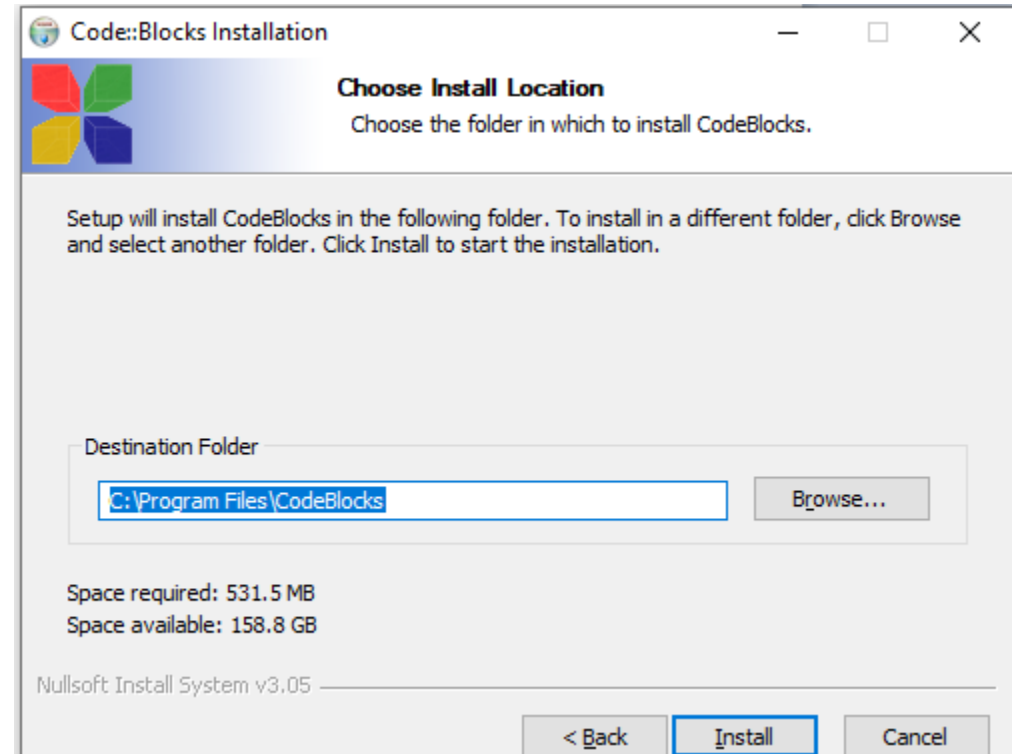
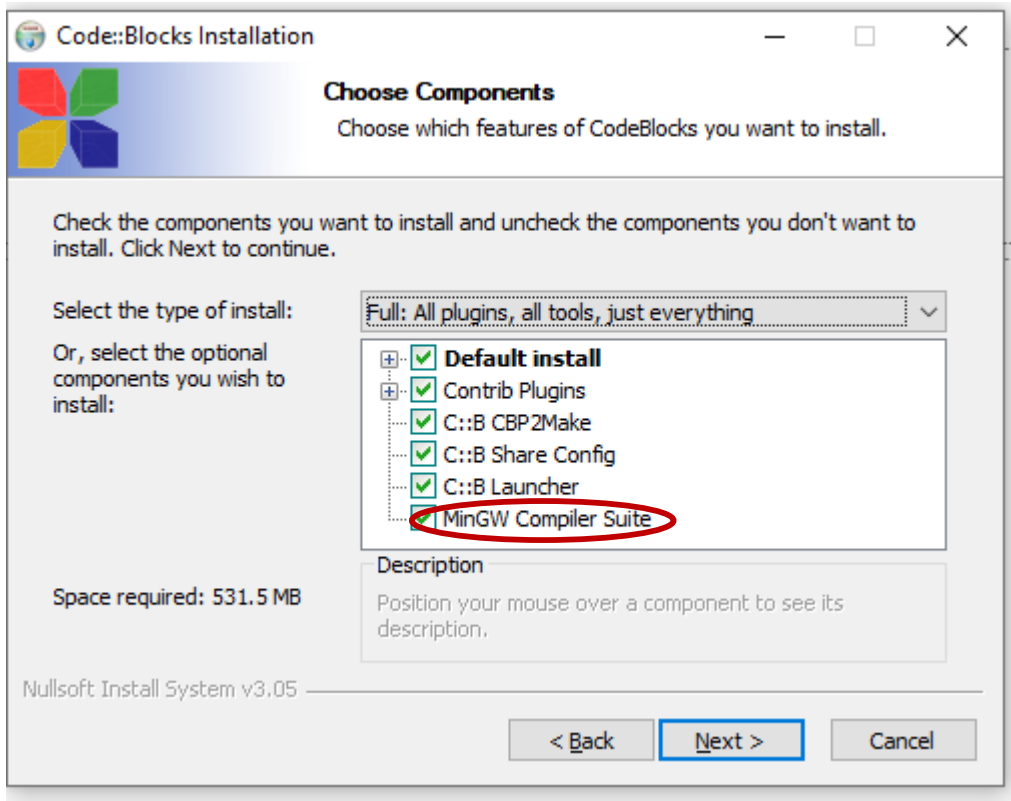
The screenshot shows the codeblocks.org/downloads/26 page. The main content area has a heading "Please select a setup package depending on your platform:" followed by a list of options: "Windows XP / Vista / 7 / 8.x / 10", "Linux 32 and 64 bit", and "Mac OS X". Below this is a table of download links. The table has three columns: "File", "Date", and "Download from". The row for "codeblocks-20.03mingw-setup.exe" is circled in red, and the "Download from" column for that row is also circled in red. A red arrow points from a text box on the right to this circled area.

File	Date	Download from
codeblocks-20.03-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	02 Apr 2020	FossHUB or Sourceforge.net

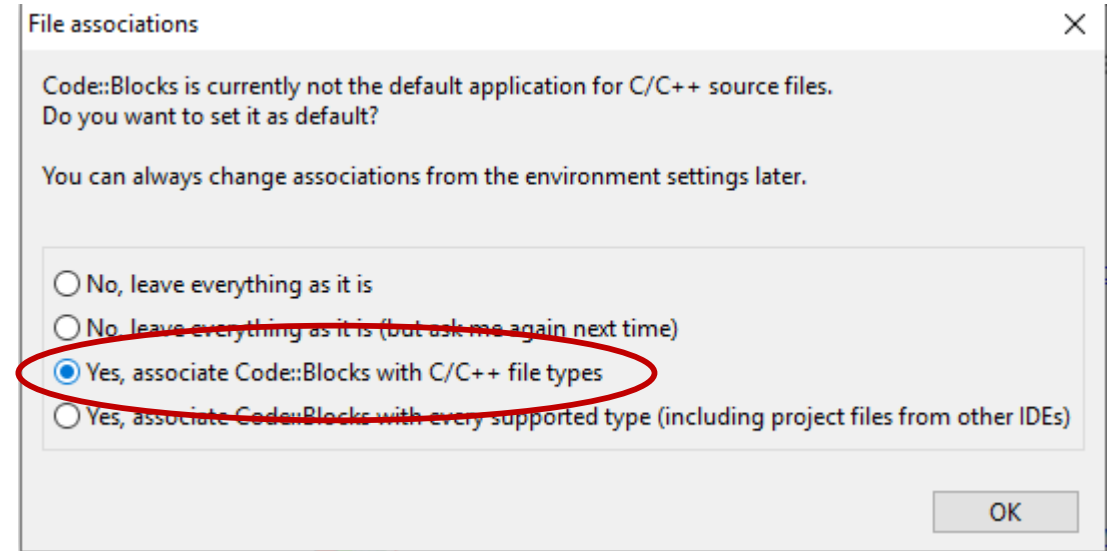
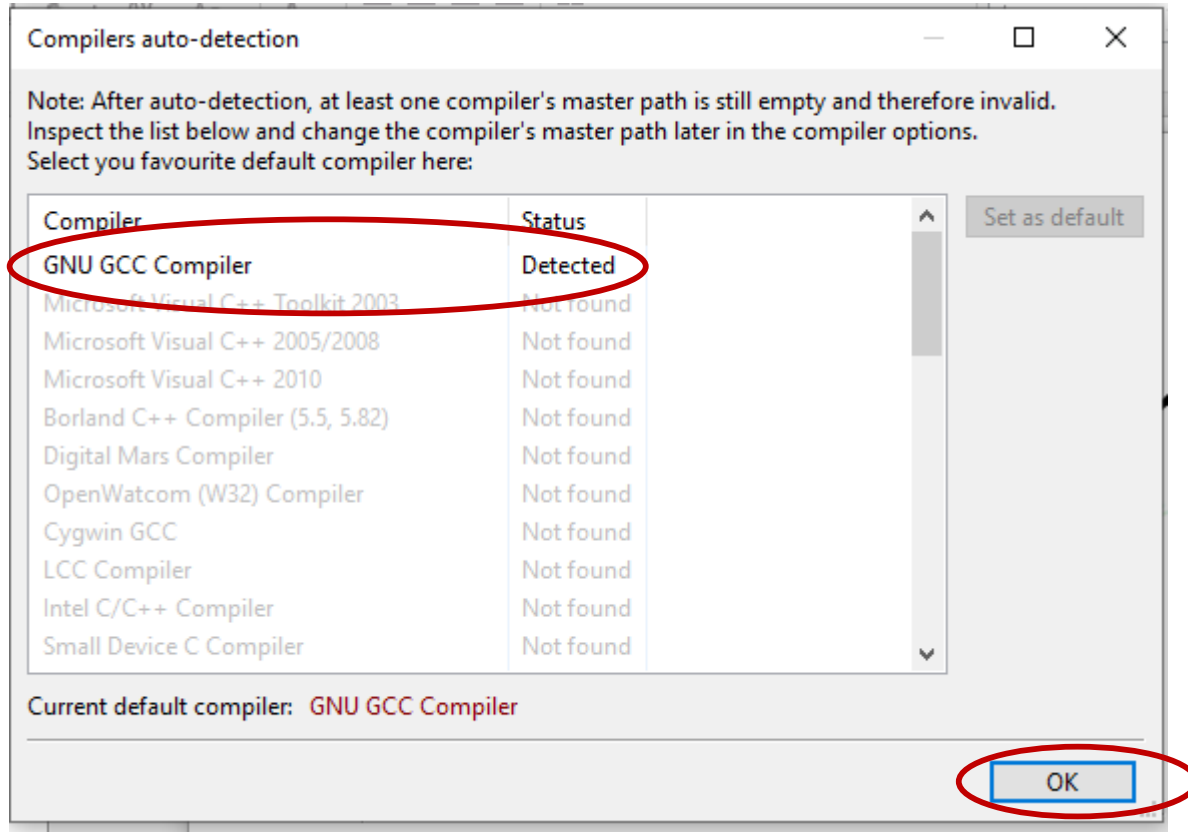
Select the site to download from.
From FossHUB the download started by default after clicking FossHUB.

Install run the executable file you downloaded, agree to the terms.

When installing it, double-check that the MinGW compiler is checked/included



We will use the GNU GCC Compiler.
(other C compilers should also be ok, but if available,
choose the GNU one for consistency.)



Check that you can “Build and run” a file. Follow the next pages to create a file and try to “Build and run” it. If it does not work, it could be because Code::Blocks cannot find the C compiler. See page 13 (after the section on creating and running a C file) that shows how to make Code::Blocks.

Create a C file, compile it and run it.

Project or no project?

- Larger pieces of code consist of multiple files and are developed using a project that organizes all those files.
- You do NOT need to create a project for now. We will do that later on so that we can Debug our code. (Method 2 below will be used to add the file to a Project in order to debug it.)
- We will create just a C file.

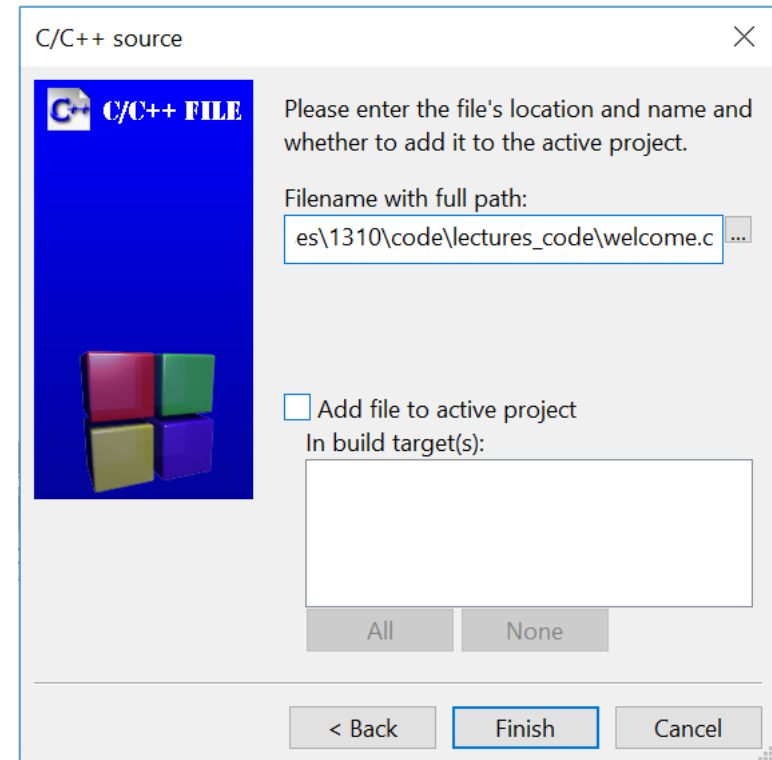
Create an empty C file

Method 1:

- File -> New -> Empty file (This will create a file called Untitled)
- File->Save file as...
- Navigate to the location where you will store your code from this class (e.g. \courses\1310\code\lectures_code) and enter the desired file name: welcome.c

Method 2:

- File -> New -> File ...
- Select *C/C++ source* and click *Go*
- Select *C*
- Navigate to the location where you will store your code from this class (e.g. \courses\1310\code\lectures_code) and enter the desired file name: *welcome.c* and click *Save*
- Do NOT select *Add file to active project* click *Finish* (Later we will need to add the file to a project in order to be able to debug it, but for now, we will just create a file.)

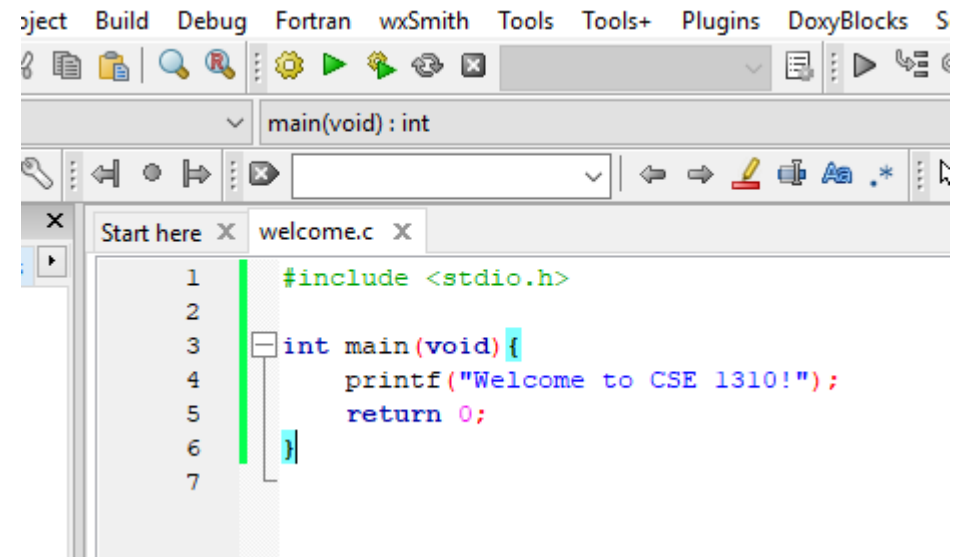


Write code in the C file

Type the text below in the file (the text will be colored):

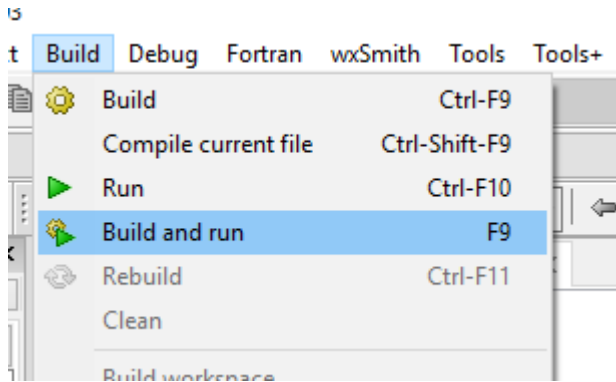
```
#include <stdio.h>

int main(void)
{
    printf("Welcome to CSE 1310!");
    return 0;
}
```

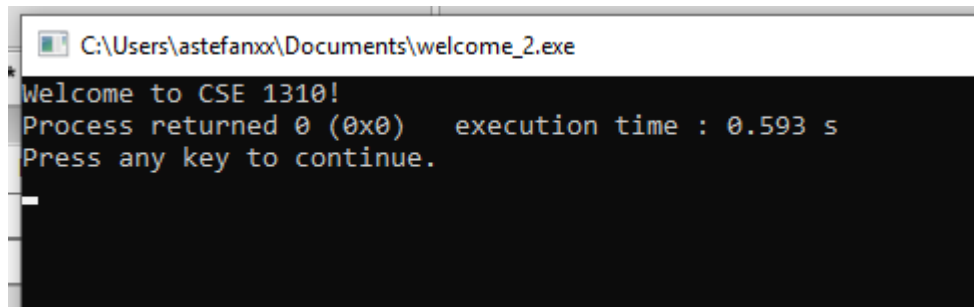


Select: Build->”*Build and run*”

Note for future: make sure every time you need to compile and run your code you select “*Build and run*”, not just “*Run*”. This way the new (most recent) code is compiled, as opposed to running the previously compiled code. It is similar to refreshing a webpage to enforce viewing the updated version.



You should see this window pop up. Notice that the first line prints what you wanted.

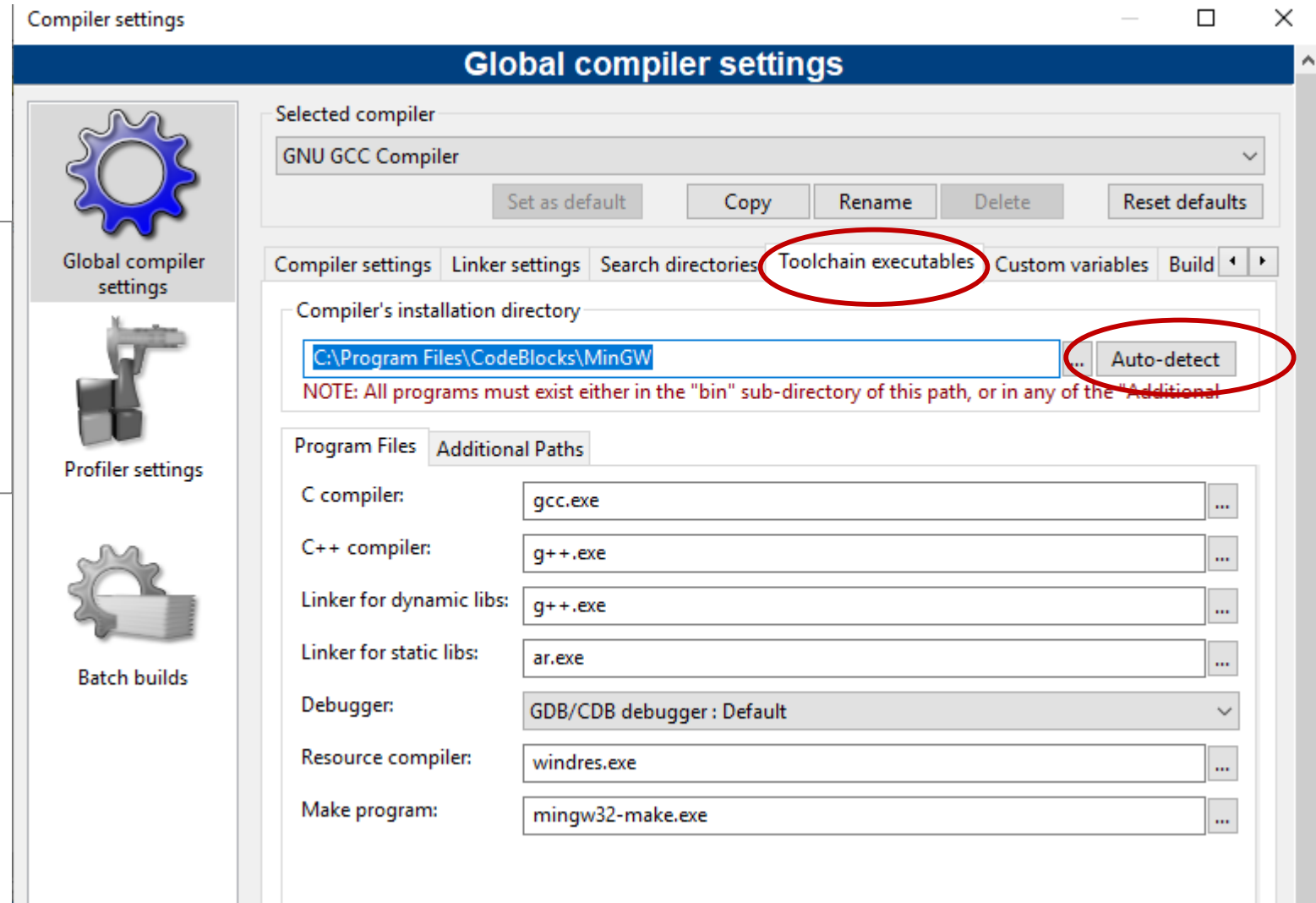
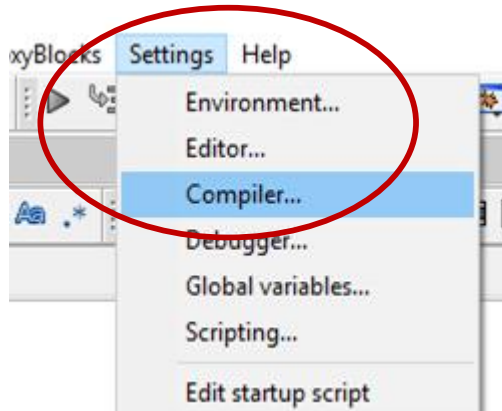


You are all set.

“Build and run” does not work

If you still can not build and run the C code, make sure the C compiler is detected by Code::Blocks. Go to:

Settings-> Compiler-> “Toolchain Executables” and click the “Auto-detect” box



If it could not auto-detect the compiler:

- If you DO know where the compiler is, navigate to that location in the “Compiler’s installation directory” box
- If you do NOT know where the compiler is, there may not be a compiler on your machine. For Mac and Unix/Linux, you need to install that separately. For Windows you can also install the compiler separately, but it is simpler to uninstall Code::Blocks and then make sure you download the version that has the compiler (MinGW) together with Code::Blocks (e.g. codeblocks-20.03-setup.exe). See slide 5.

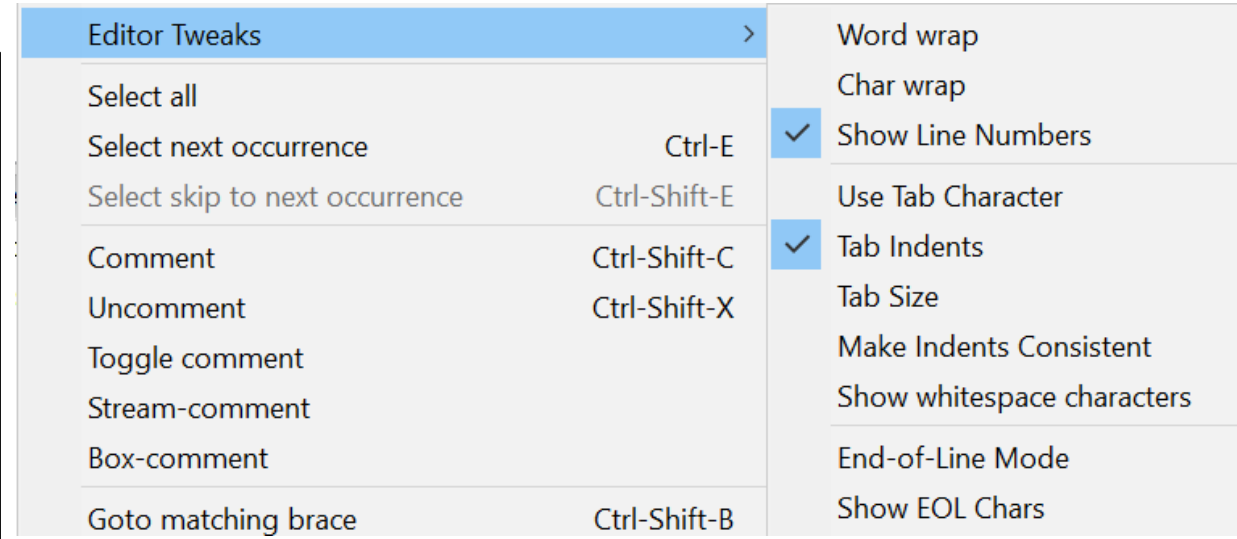
Other problems during installation?

- There should not be any problems during installation, but if you do get some error messages, search the web. Include the error message and your system (e.g. Windows 10).

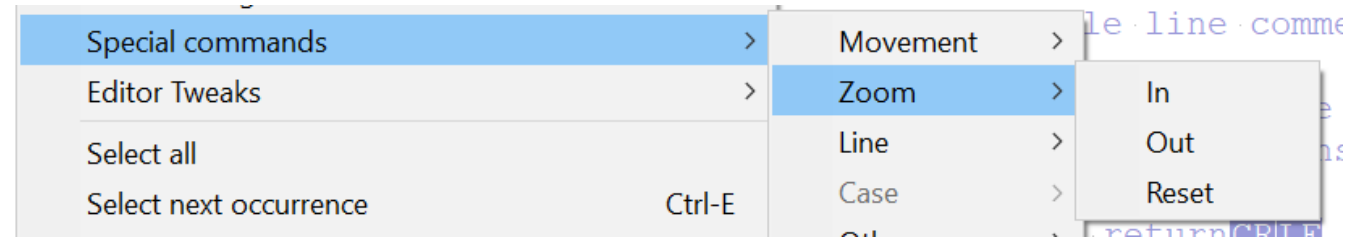
Code::Blocks Settings

To show line numbers and change other settings go to: Edit -> "Editor Tweaks"

- And then "Show Line Numbers"
- Also note the "Show EOL Chars". EOL stands for End Of Line. Files from different Operating Systems will have different EOL characters: Windows - CRLF, Mac – CR, Unix – LF . See also the table from the Wikipedia page <https://en.wikipedia.org/wiki/Newline>.



To change font size in Editor window:
Edit -> "Special commands" -> Zoom -> In/Out

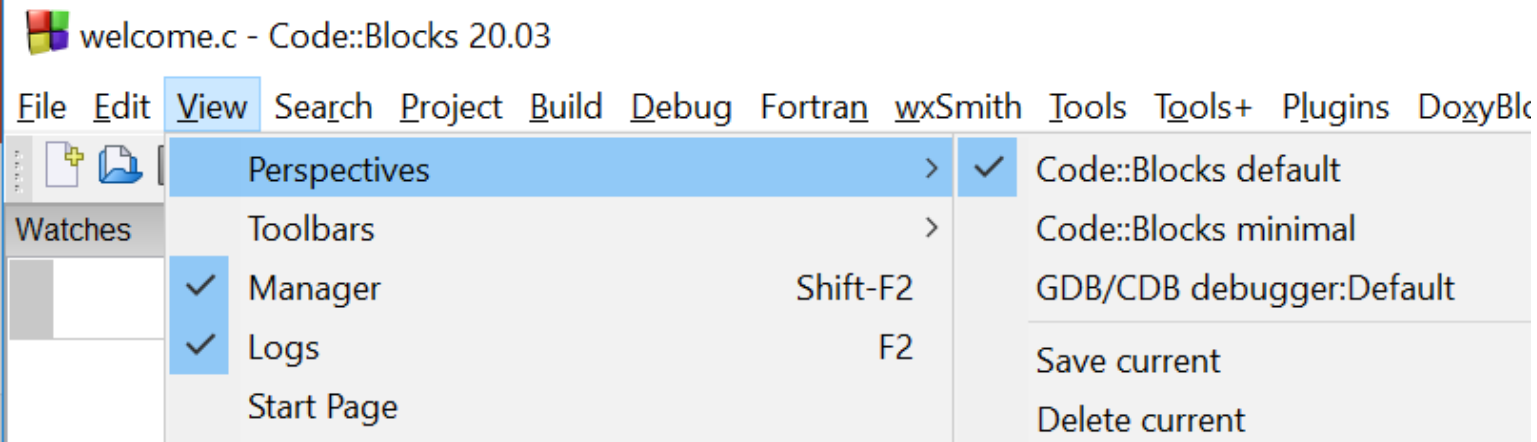


To indent a group of instructions: select the instructions and then click:

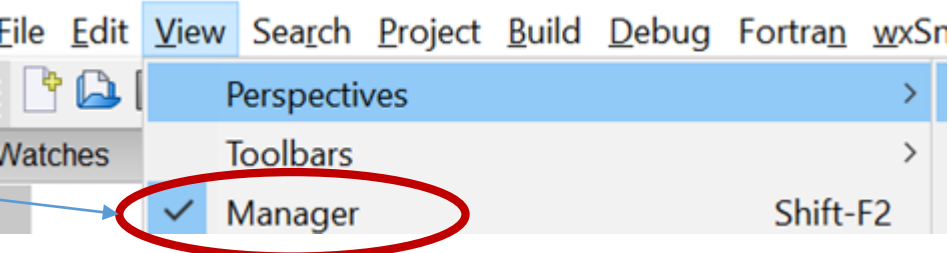
- TAB – to indent
- Shift + TAB – to decrease the indent

Code::Blocks Settings

To restore the original perspective or select other perspectives:
View -> Perspectives
You would need this if you closed some windows (e.g. Debugger, Build Messages) and want them to be visible again.
You can also create your own perspective.

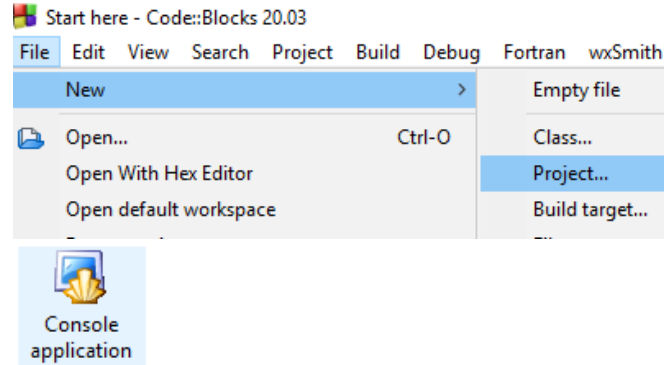


If you cannot see the Projects Panel:
View and check the Manager



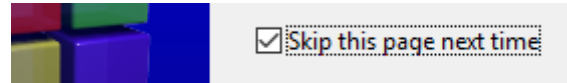
Create a project

- File -> New -> Project

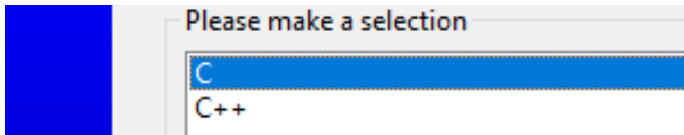


- Select: “Console Application” and click Next

- Check the “Skip this page next time” - You will not see this welcome page next time



- Select C for the language to use.



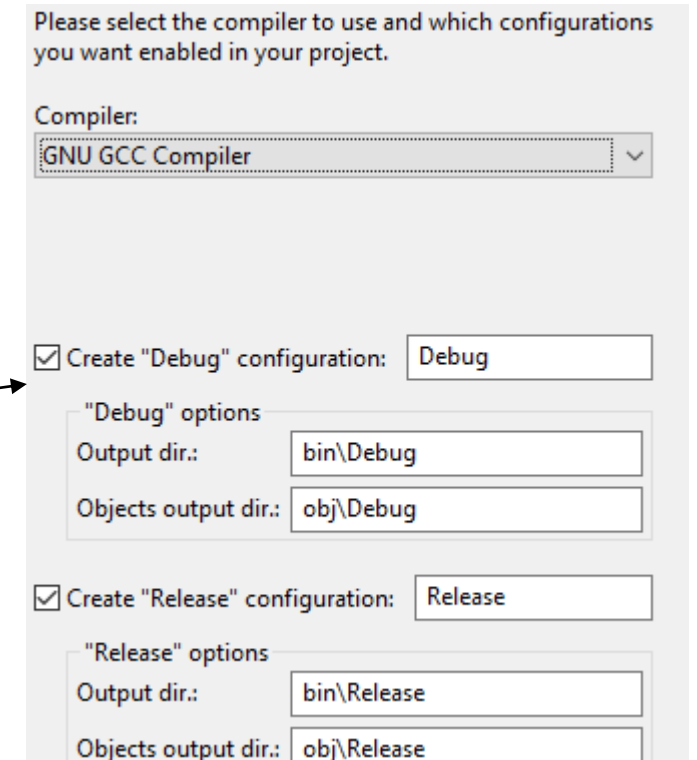
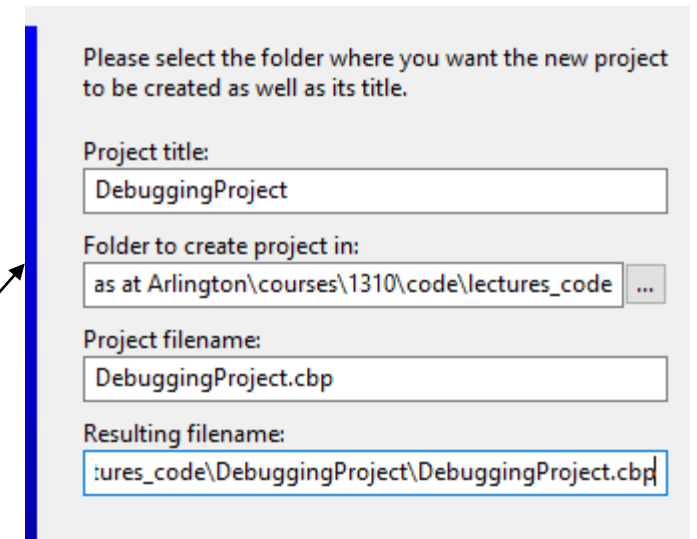
Click *Go*

- Give the project a title (e.g. DebuggingProject) and select a folder to place it in (e.g. lecture_code)

- The default configurations should work. Double check that the *Debug* and *Release* check-boxes are checked

Click *Finish*


- Done. You will see the project. (continue to next page) (in the Management panel under the Projects view)

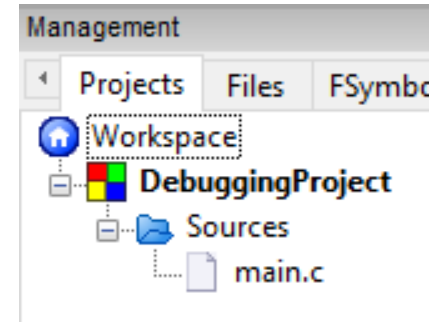


Using a project

- You will see the project in the IDE.

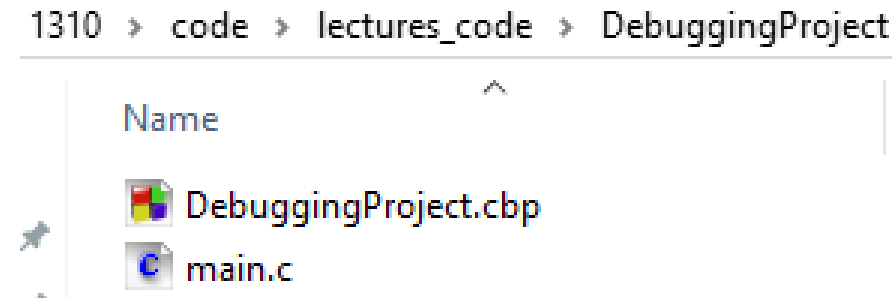
(It is in the *Management* panel under the *Projects* tab.)

- By default a file called main.c was created in the project. To see it, click + next to *Sources*
- Double-click on main.c and it will open in the editor. It already contains starter code to print *hello world!*
- Click *Build and Run*  and see the program output
- You can edit the file main.c (write any valid C code in it) and run and DEBUG that code.



Where are the project files?

- Use a file explorer program and navigate to the folder where you created the project (e.g. 1310/code/lectures_code). You will find a folder with the project name and inside there the project file (with extension .cbp) and the main.c file
- Note: If you are using a different IDE, the location of the files and the project file extension (.cbp) will be different.

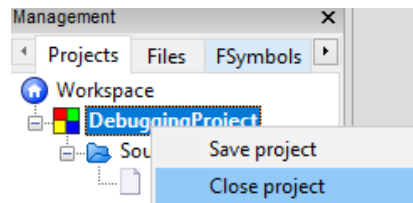


Using a project

- A project is a way to organize multiple files needed for a larger application.
- You can close and reopen the project. This is convenient when you have multiple projects.

Close a project

- *When you close the project it simply does not SHOW in the panel. All the code is still on your computer.*
- To close a project right click on it and select *Close Project*



Open a project

- When you open a project, it makes it visible in Code::Blocks and you can run and debug the code in that project from Code::Blocks.
- To open go to: File-> Open and then navigate to where the .cbp file is for the project you want to open

Project: add/remove files

You can **add or remove files from a project.**

Remove a file from a project

- To remove a file: right-click on it and select *Remove file...*
- If you remove a file from a project *it does NOT delete* the file, it simply removes it from the IDE's list of files associated with that project.

Add a file to a project

- To add an existing file:
right-click on the project name
select *Add files ...* and
navigate to the file you want to add
- To create a new file and make it part of the project:
Make sure this project is active (it is listed and its name is in bold font. If not, right-click on it and select *Activate Project*)
Go to *File -> New -> Empty file*
Click *Yes* to add this new file in the active project
Enter the file name
Double-check that the *Debug* and *Release* boxes are checked and click *Ok*

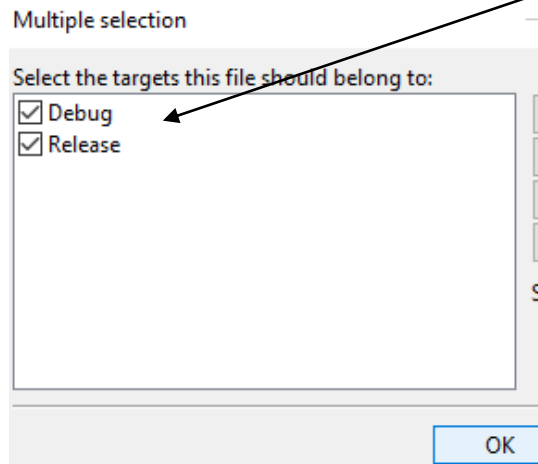
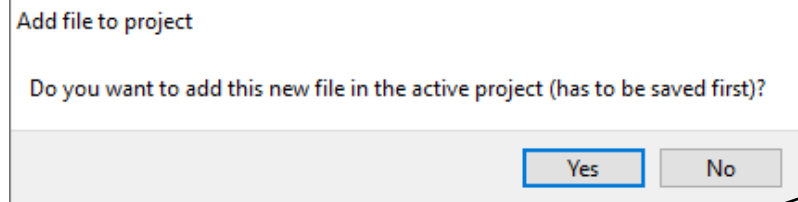
There are other ways to add files or rename files from a project.

There can be many .c files part of a project but ONLY ONE

```
int main(void)    function.
```

If multiple .c files belong to a project, only one file can have a `main` function in it.

For our class, you only need to have one .c file part of a project.



Using a project to Debug code

- Debug - The IDE provides a convenient framework to examine what is happening with the program (what line is being executed at any time and the value of the data in the program at that time). This is done in DEBUG mode.
- In order to debug a program, it must be part of a project. (We can run an individual file, but we can only debug a project)

- You have 2 options:

Option 1: Create a project for every program you need to run. If that is for a homework, make sure you name the .c file with the required name. The name of the project can be whatever you want.

Option 2: Create one project that will be used for debugging only. When working on a program, add it to the Debugging project and debug it as needed. Remove it from the project when you are done.

- You can have several .c file in a project, but only one of those files can have a main() function in it (because that is where the program execution will start and it cannot have multiple start points.)

Running individual files vs projects

- When you Build and Run... it will be done for the ACTIVE PROJECT.
- If you have multiple projects, make sure the one you want to work on is active. (You can close the others, or explicitly make this one the active one – right click on the project name to get that option)
- If you want to run a single file, then ALL your projects must be closed (there should not be any project in the list of projects)