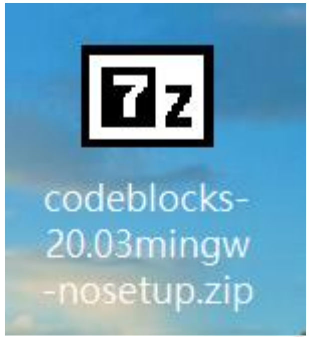
CodeBlocks tutorial

Install

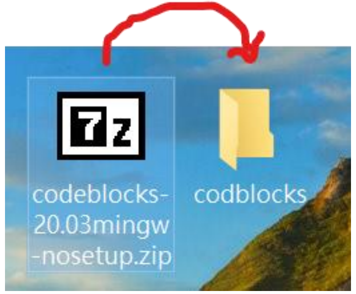
- We will use codeblocks 20.03 with mingw nonsetup version
- Note: we highly recommend you using codeblocks and with the right mingw version, otherwise unexpected error will occur in your final project.
- For MAC user, we recommend Visual Studio Code
- Windows: https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-nosetup.zip/download
- MAC: Visual Studio Code - Code Editing. Redefined
- Linux: https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Linux

Basic

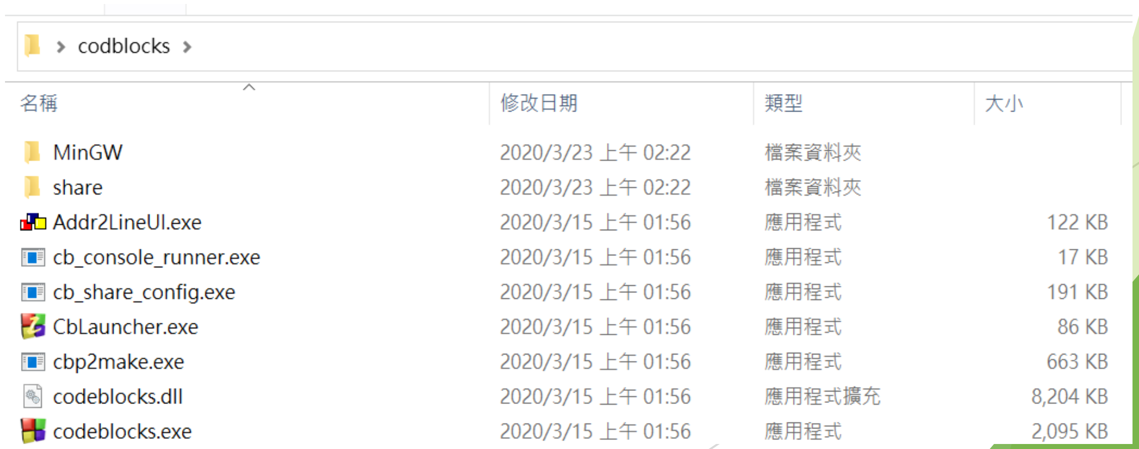
- You will got something like this



- Create a folder and the extract in it



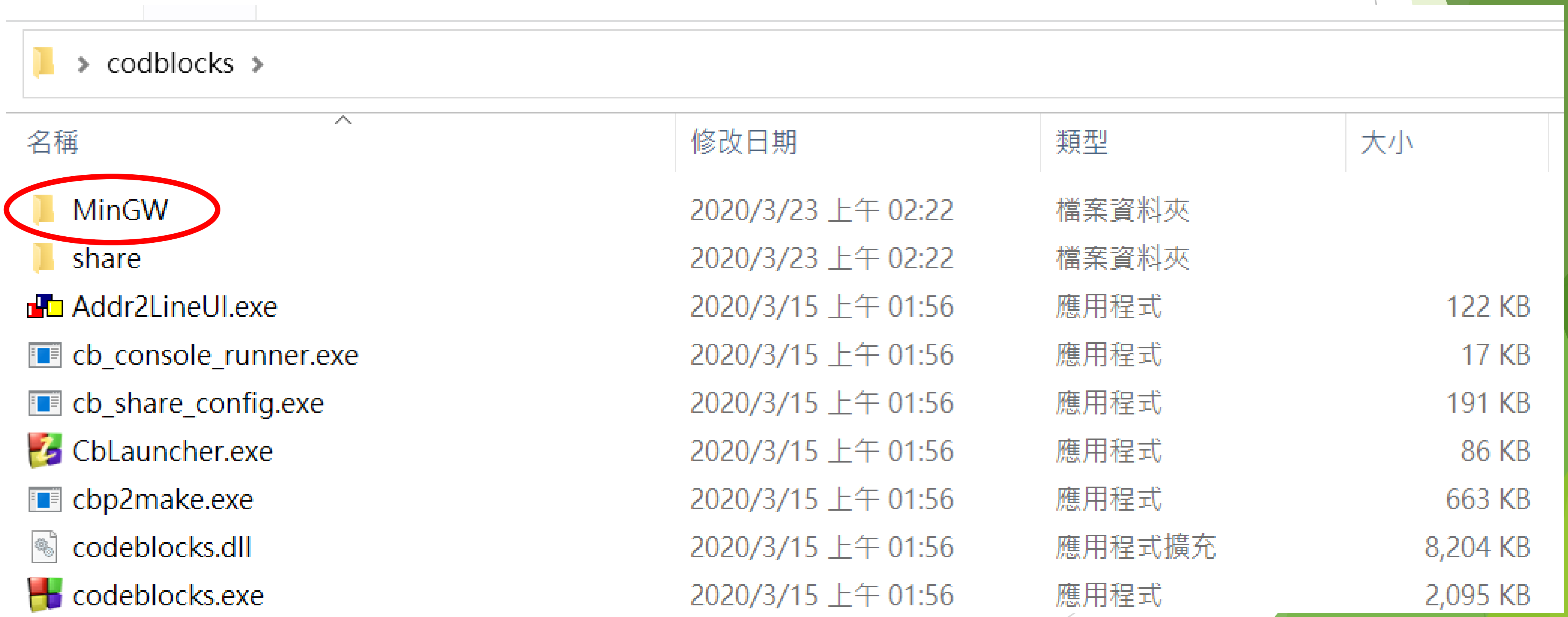
- You will got something like this



名稱	修改日期	類型	大小
MinGW	2020/3/23 上午 02:22	檔案資料夾	
share	2020/3/23 上午 02:22	檔案資料夾	
Addr2LineUI.exe	2020/3/15 上午 01:56	應用程式	122 KB
cb_console_runner.exe	2020/3/15 上午 01:56	應用程式	17 KB
cb_share_config.exe	2020/3/15 上午 01:56	應用程式	191 KB
CbLauncher.exe	2020/3/15 上午 01:56	應用程式	86 KB
cbp2make.exe	2020/3/15 上午 01:56	應用程式	663 KB
codeblocks.dll	2020/3/15 上午 01:56	應用程式擴充	8,204 KB
codeblocks.exe	2020/3/15 上午 01:56	應用程式	2,095 KB

Basic

- Make sure you have MinGW folder with your codeblocks

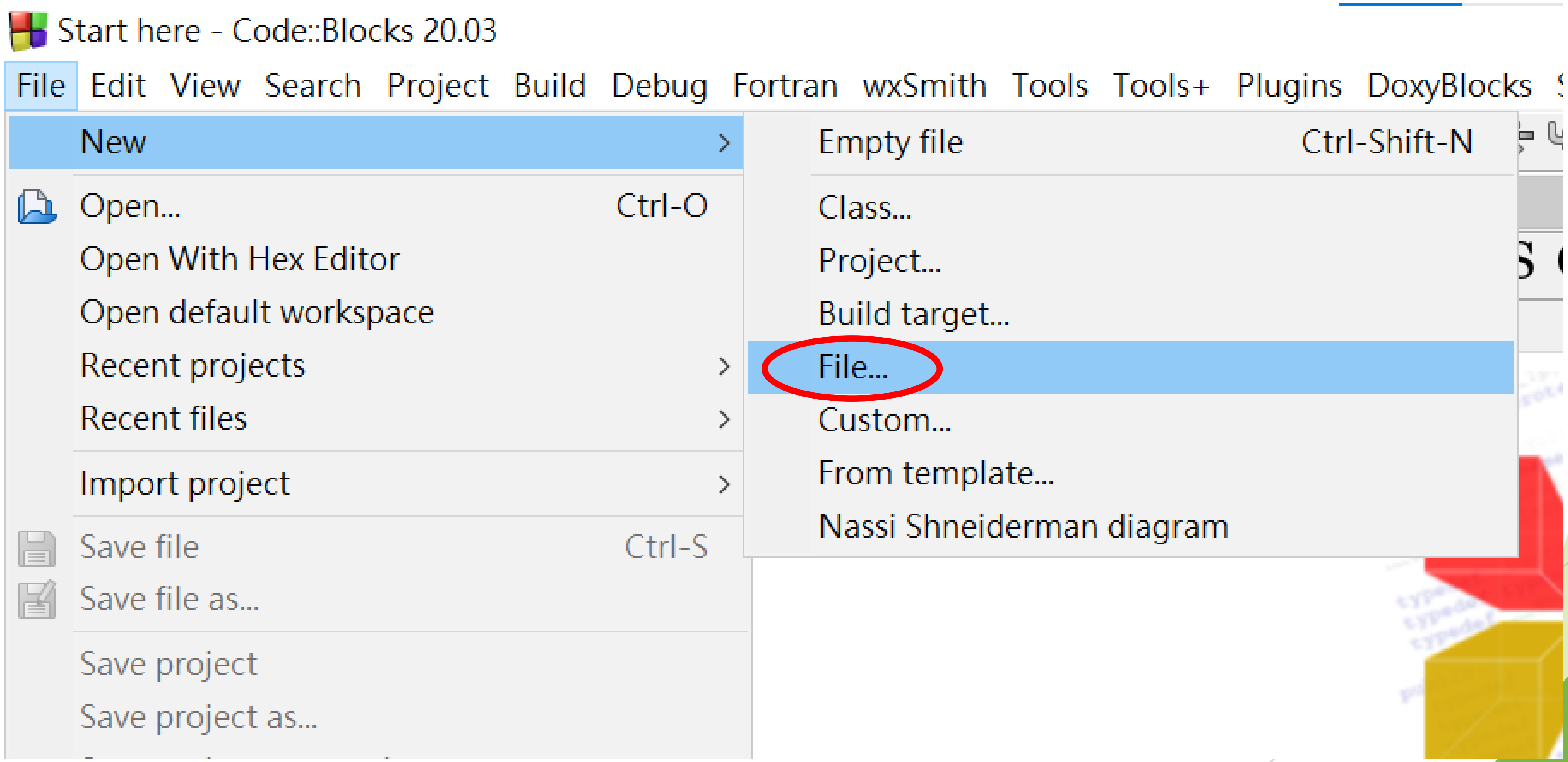


› codblocks ›

名稱	修改日期	類型	大小
MinGW	2020/3/23 上午 02:22	檔案資料夾	
share	2020/3/23 上午 02:22	檔案資料夾	
Addr2LineUI.exe	2020/3/15 上午 01:56	應用程式	122 KB
cb_console_runner.exe	2020/3/15 上午 01:56	應用程式	17 KB
cb_share_config.exe	2020/3/15 上午 01:56	應用程式	191 KB
CbLauncher.exe	2020/3/15 上午 01:56	應用程式	86 KB
cbp2make.exe	2020/3/15 上午 01:56	應用程式	663 KB
codeblocks.dll	2020/3/15 上午 01:56	應用程式擴充	8,204 KB
codeblocks.exe	2020/3/15 上午 01:56	應用程式	2,095 KB

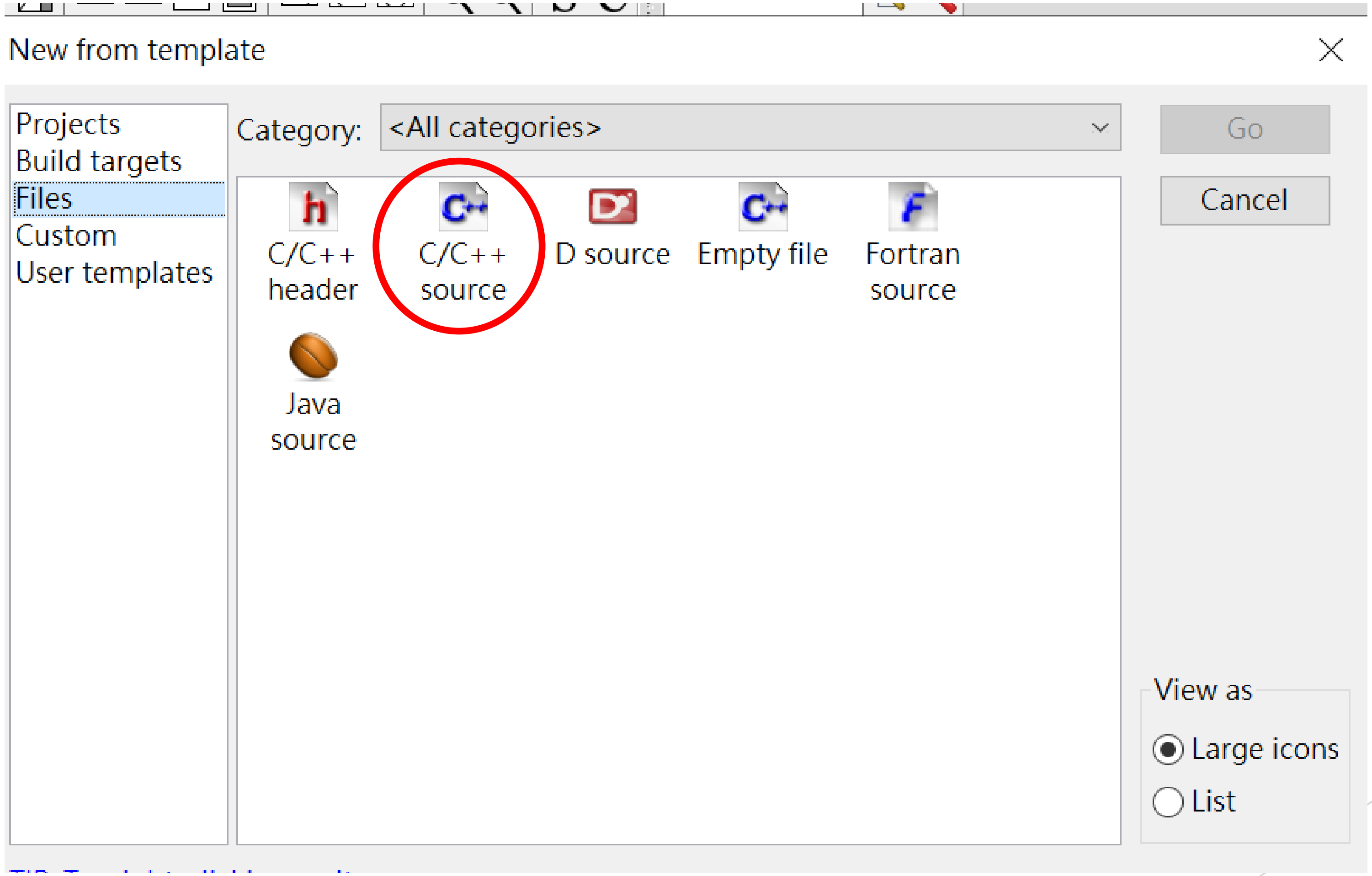
New file

- Our weekly practice you only need to create a common file, just click **File…**



New file

- It will show the interface below. Click **C/C++ source**



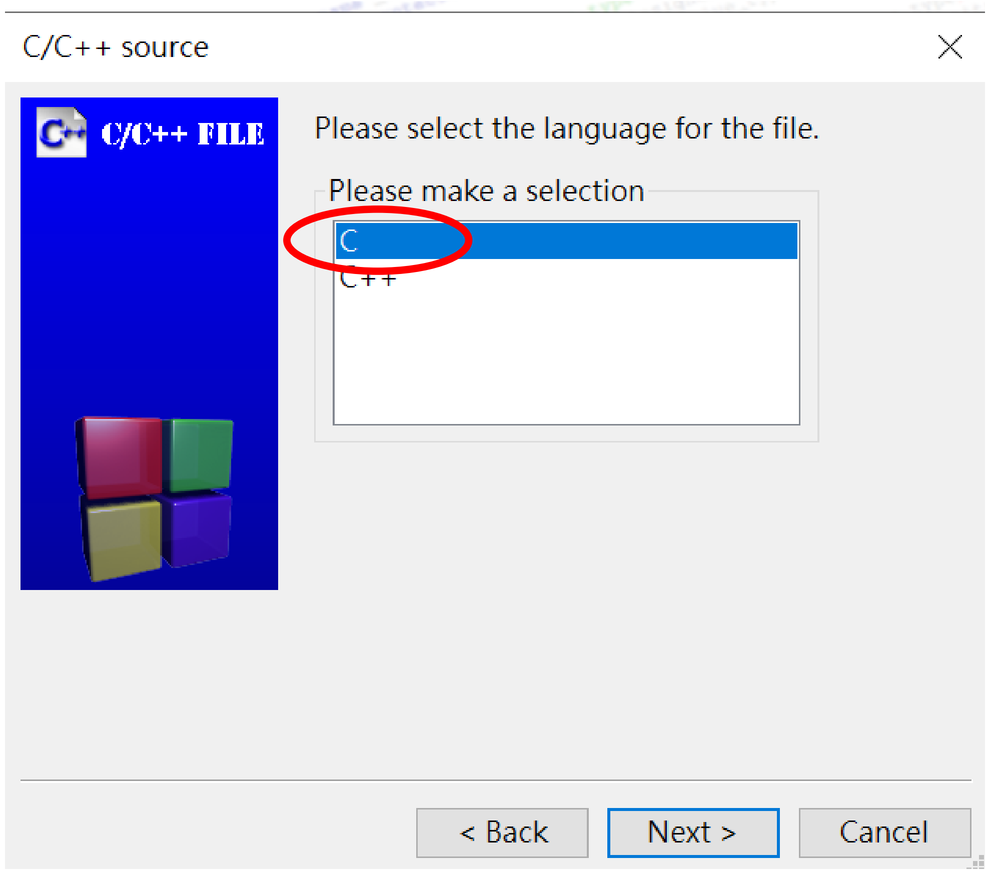
New from template

Category: <All categories>

View as
● Large icons
○ List

New file

- Click **Next** until you see the interface below, choose **C**



C/C++ source

Please select the language for the file.

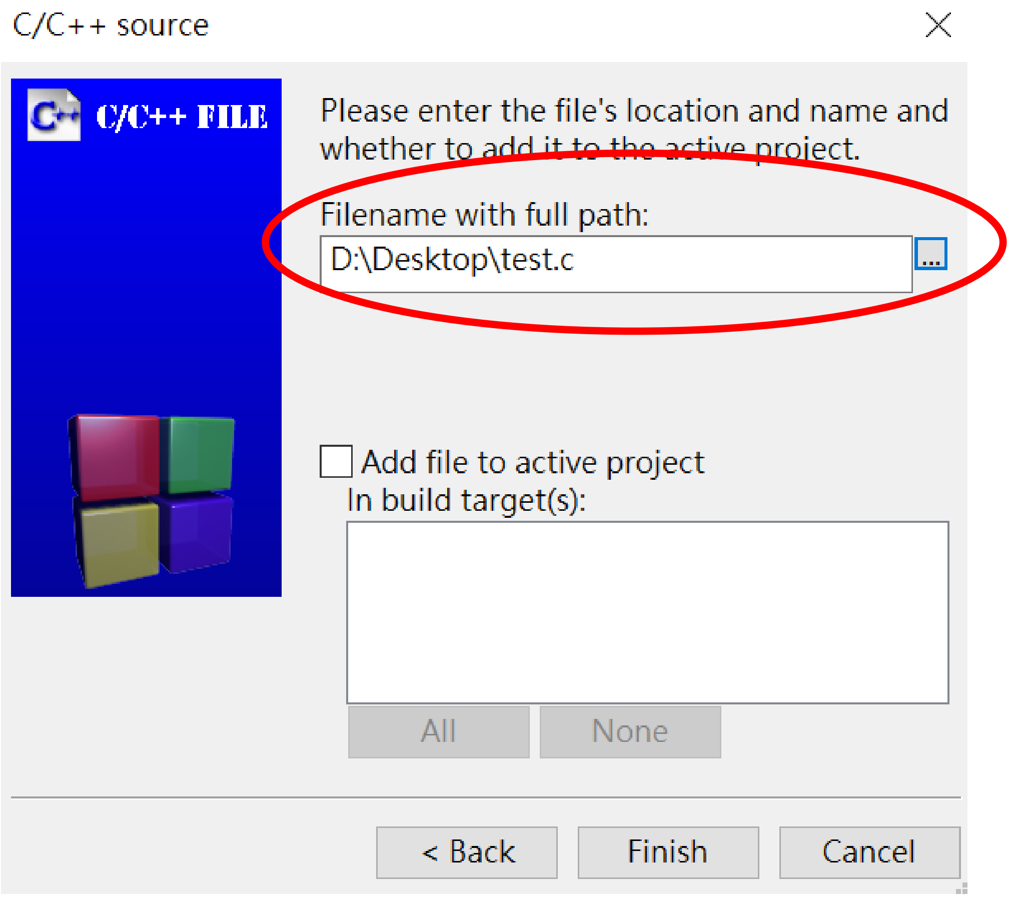
Please make a selection

C

C++

New file

- Give this file a **name** and the **path** you want to saved at then you create a **C** file!



Please enter the file's location and name and whether to add it to the active project.

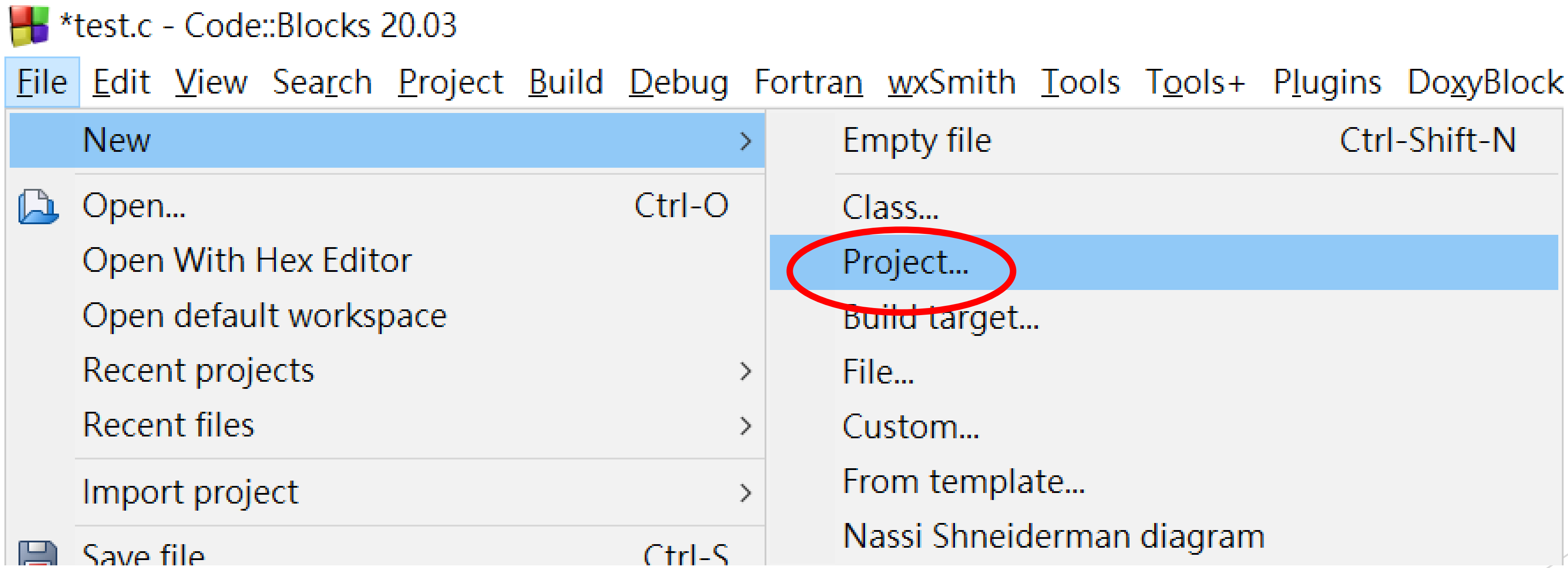
Filename with full path: D:\Desktop\test.c

□ Add file to active project

In build target(s):

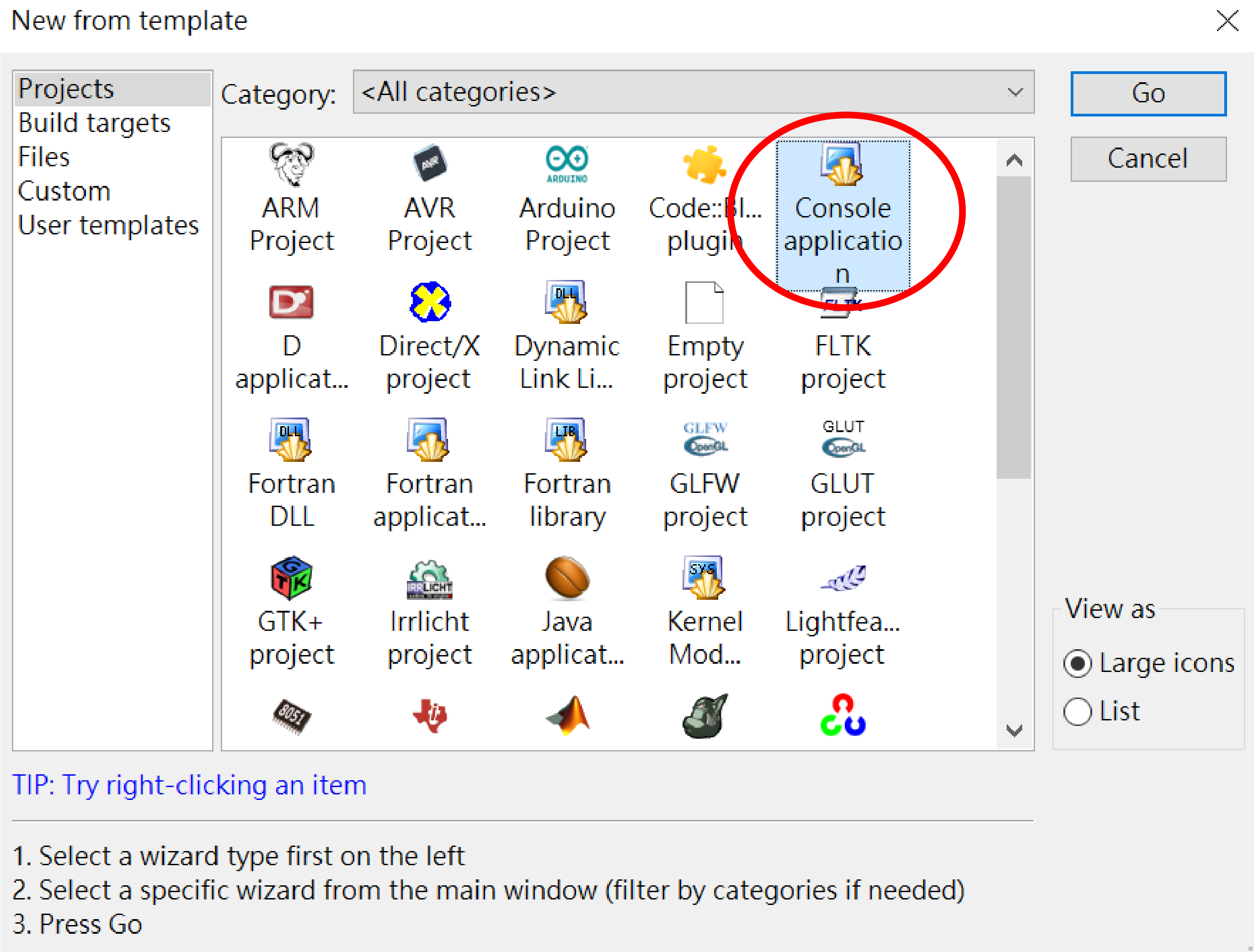
New project

- Click **Project**



New project

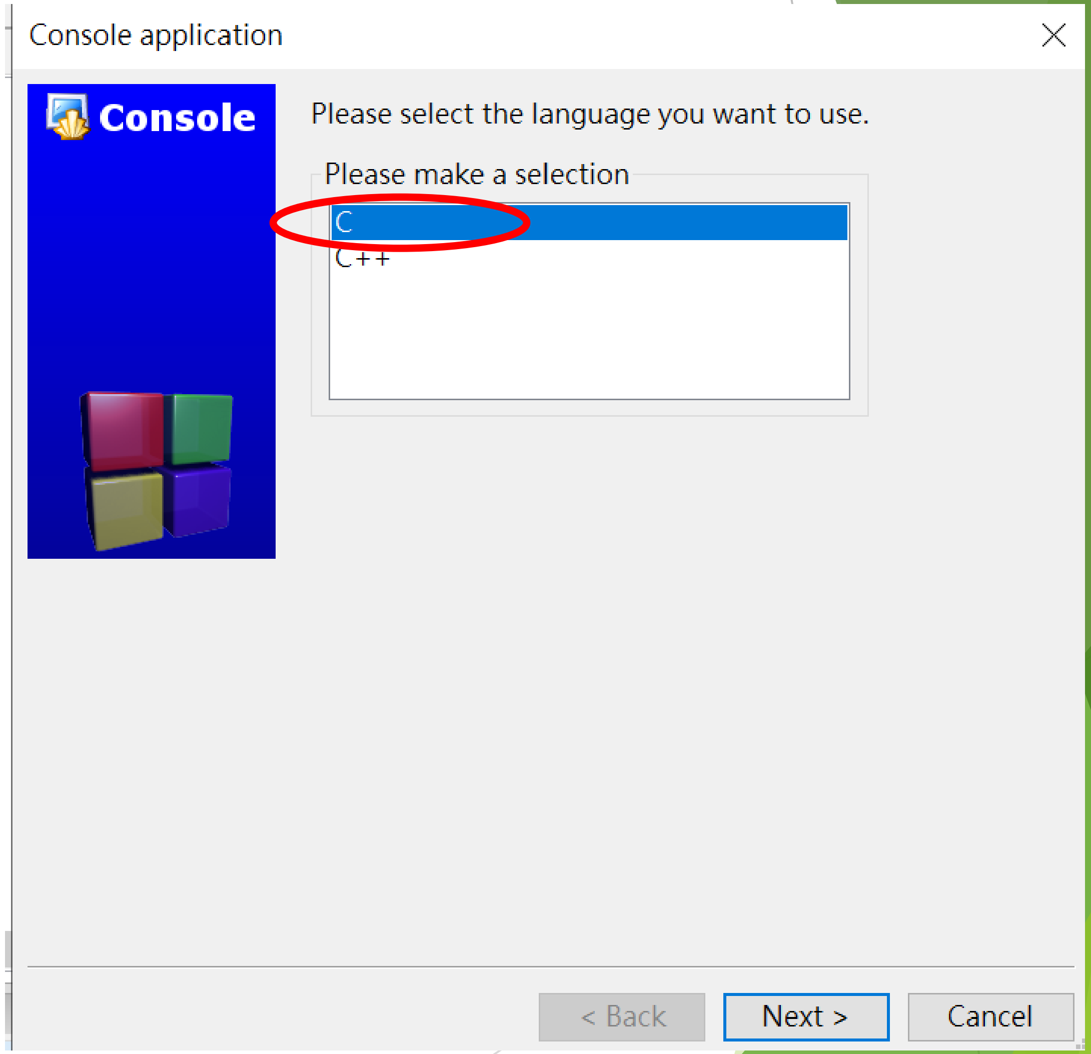
- Click **Console application**



1. Select a wizard type first on the left
2. Select a specific wizard from the main window (filter by categories if needed)
3. Press Go

New project

- Choose **C**



Console application

Please select the language you want to use.

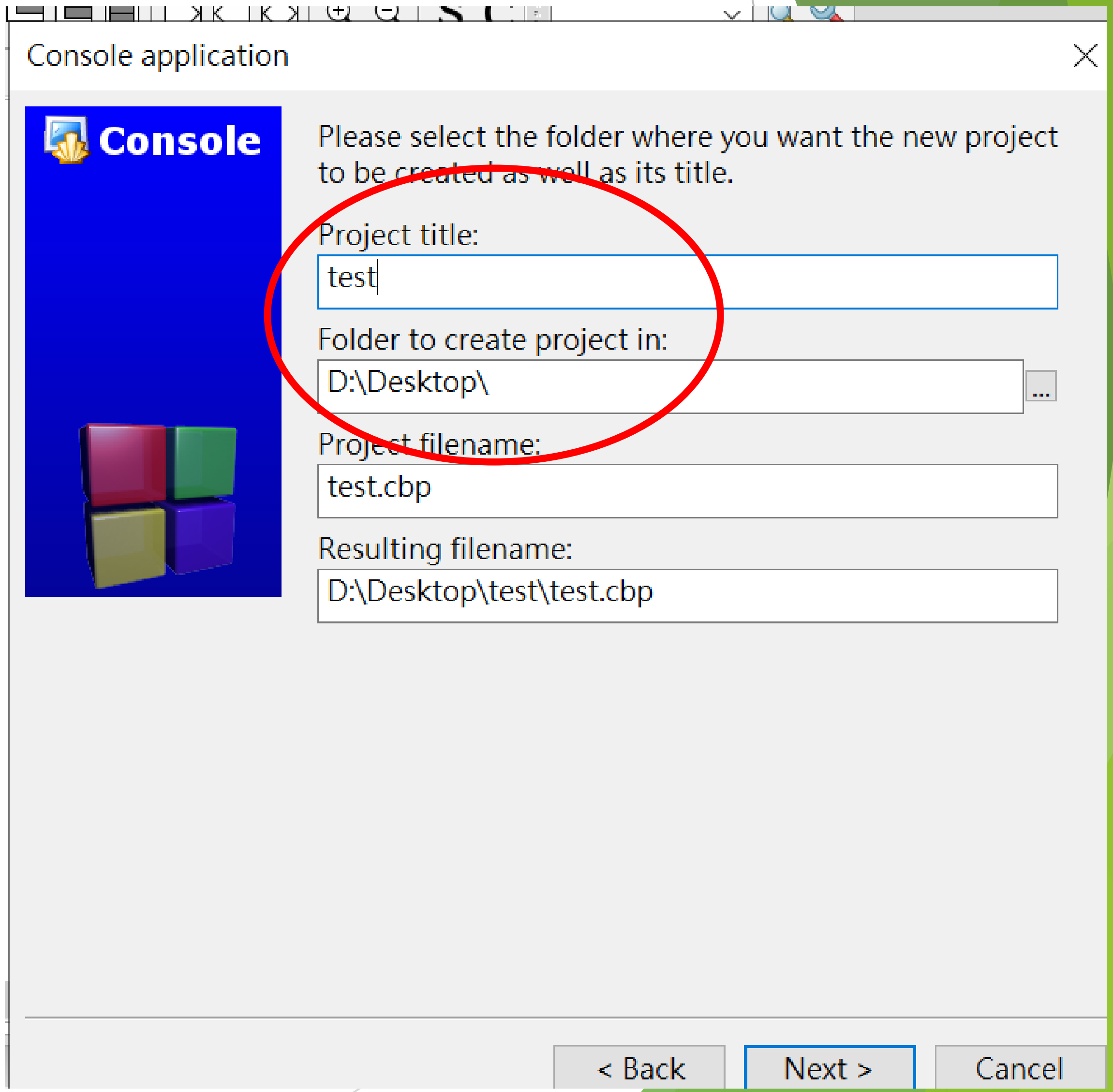
Please make a selection

C

C++

New project

- Give it a name and choose the path



Console application

Please select the folder where you want the new project to be created as well as its title.

Project title:
test

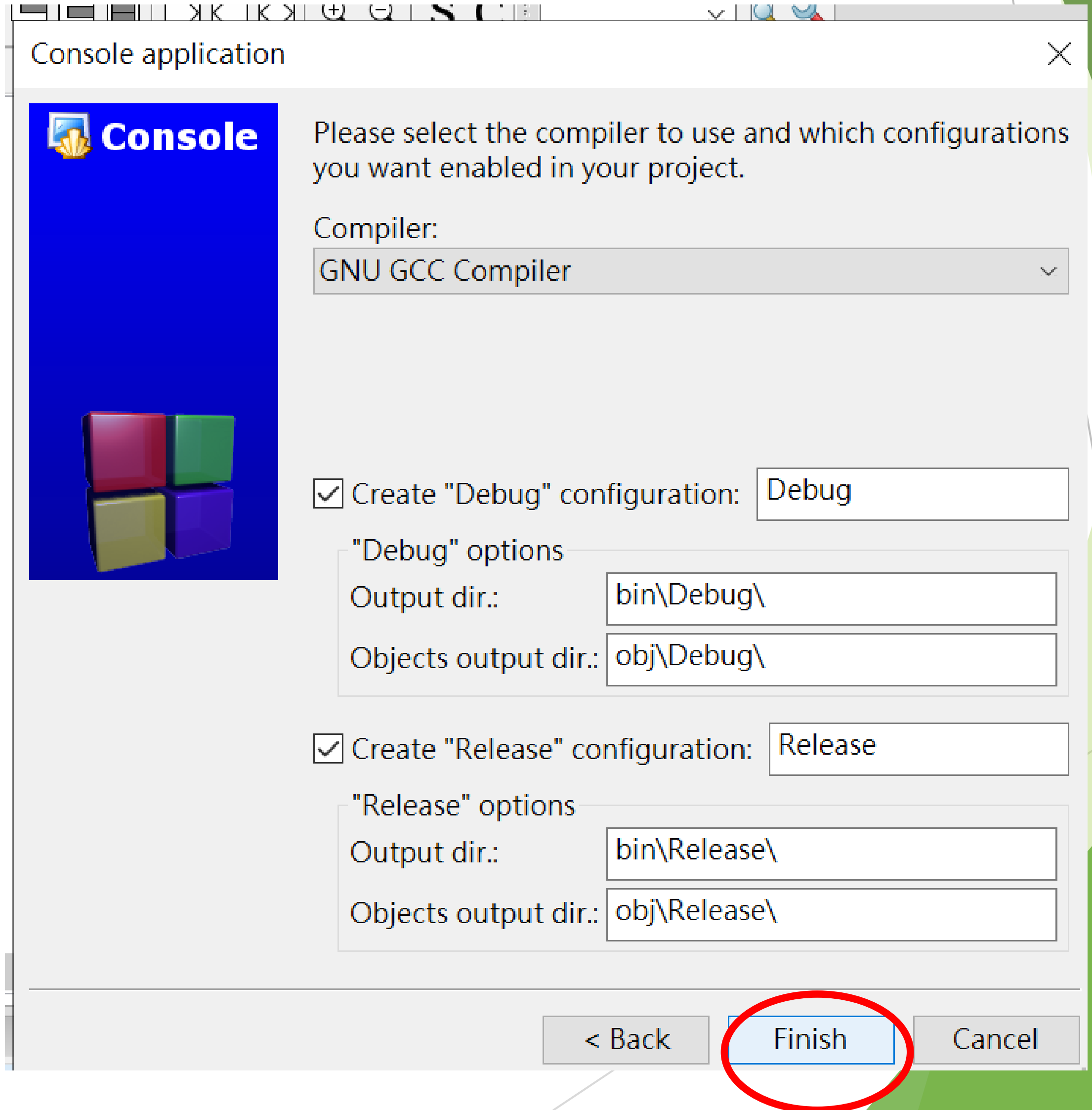
Folder to create project in:
D:\Desktop\

Project filename:
test.cbp

Resulting filename:
D:\Desktop\test\test.cbp

New project

- Just click finish



Console application

Please select the compiler to use and which configurations you want enabled in your project.

Compiler: GNU GCC Compiler

☑ Create "Debug" configuration: Debug

"Debug" options

Output dir.: bin\Debug\

Objects output dir.: obj\Debug\

☑ Create "Release" configuration: Release

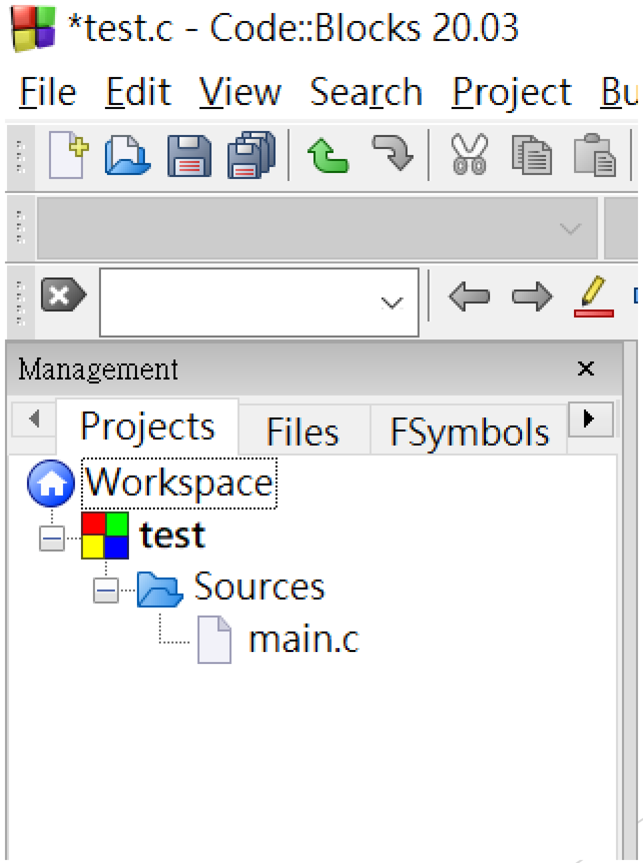
"Release" options

Output dir.: bin\Release\

Objects output dir.: obj\Release\

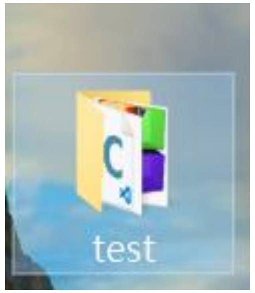
New project

- You will see something like this on your left

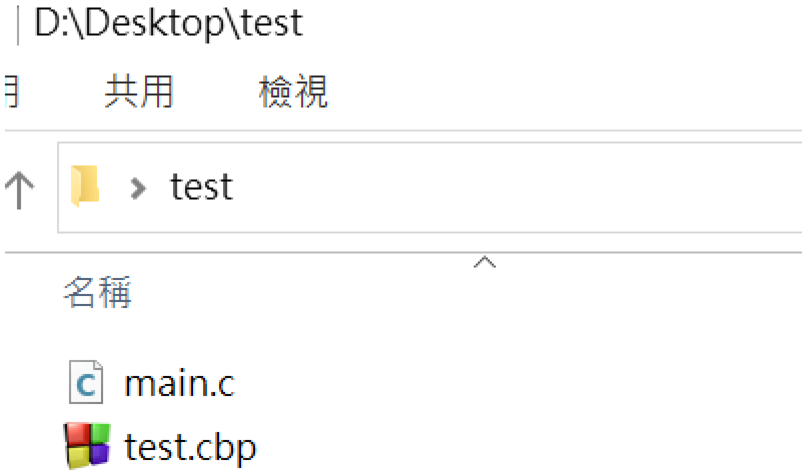


*test.c - Code::Blocks 20.03
File Edit View Search Project Bu
Management
Projects Files FSymbols
Workspace
test
Sources
main.c

- You will get something like this at the place you save



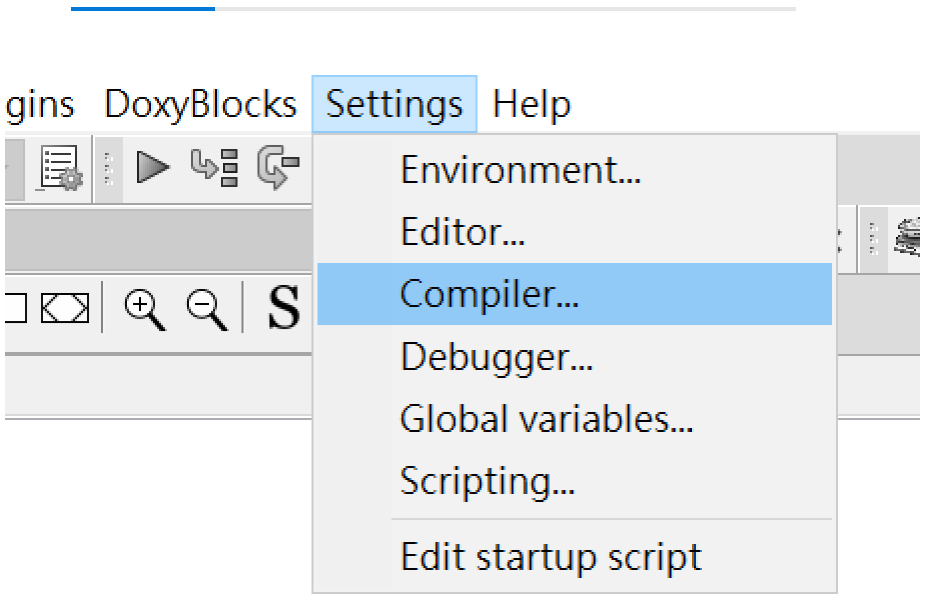
- It contains a **main.c** and ***.cbp



- **You can click ***.cbp to open your project again**

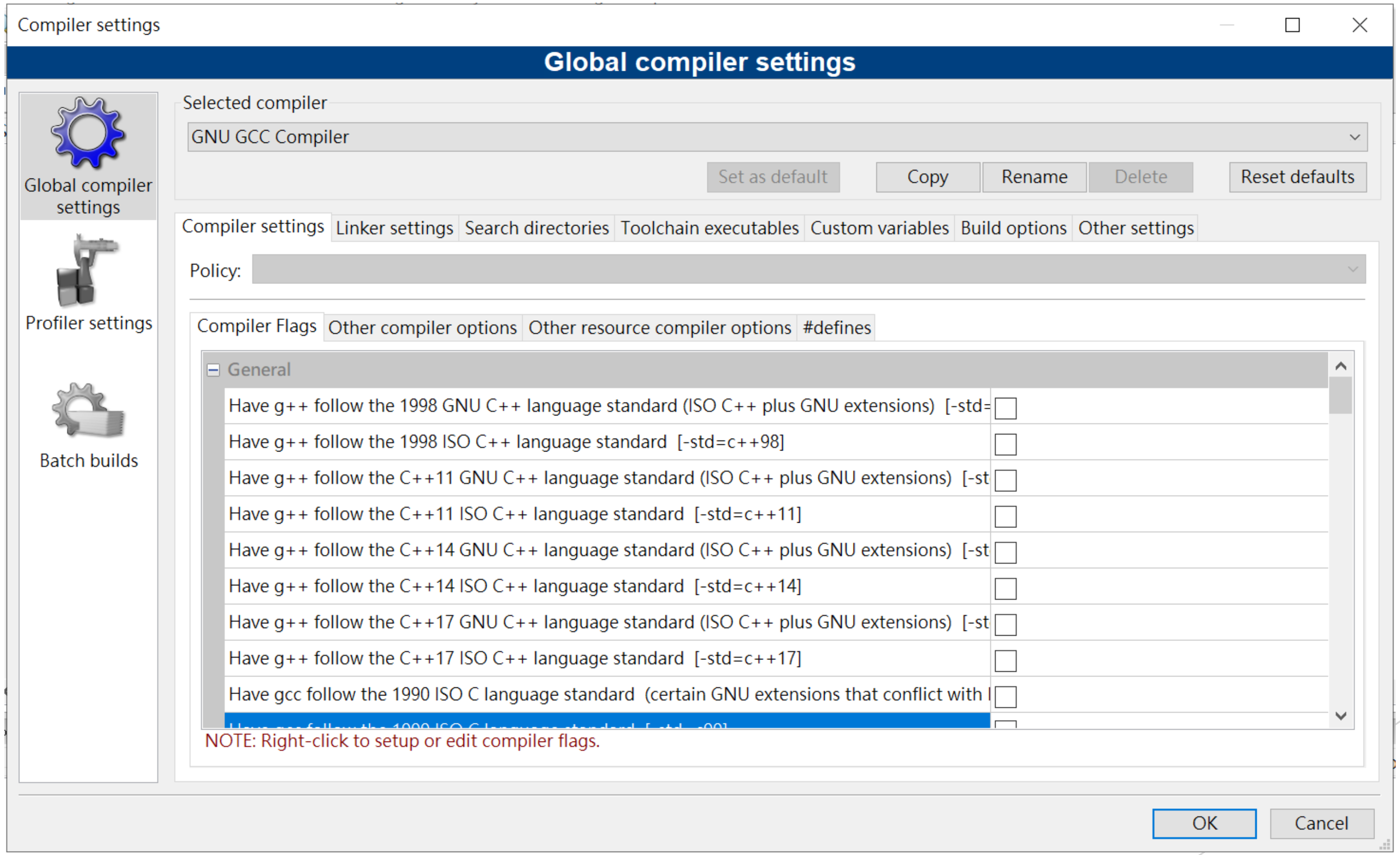
Compile

- Let’s check the Compiler



Compile

- You will enter an interface like below



Selected compiler

GNU GCC Compiler

Policy:

⊟ General

Have g++ follow the 1998 GNU C++ language standard (ISO C++ plus GNU extensions) [-std= □

Have g++ follow the 1998 ISO C++ language standard [-std=c++98] □

Have g++ follow the C++11 GNU C++ language standard (ISO C++ plus GNU extensions) [-st □

Have g++ follow the C++11 ISO C++ language standard [-std=c++11] □

Have g++ follow the C++14 GNU C++ language standard (ISO C++ plus GNU extensions) [-st □

Have g++ follow the C++14 ISO C++ language standard [-std=c++14] □

Have g++ follow the C++17 GNU C++ language standard (ISO C++ plus GNU extensions) [-st □

Have g++ follow the C++17 ISO C++ language standard [-std=c++17] □

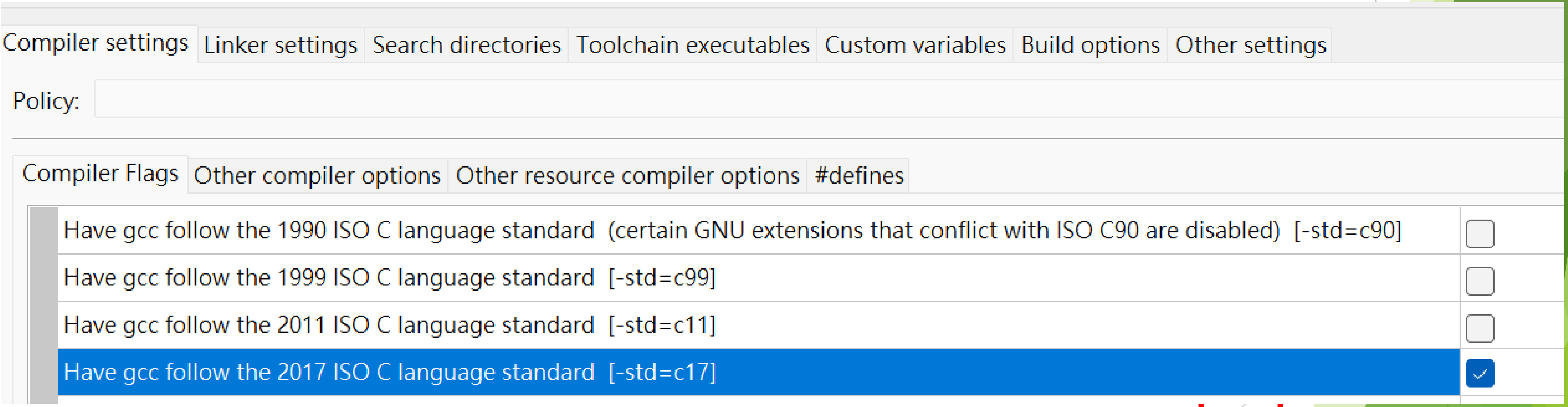
Have gcc follow the 1990 ISO C language standard (certain GNU extensions that conflict with I □

Have gcc follow the 1990 ISO C language standard [-std=c90] □

NOTE: Right-click to setup or edit compiler flags.

Compile

- Make sure you check the checkbox of **c99 or c11 or c17**



Have gcc follow the 1990 ISO C language standard (certain GNU extensions that conflict with ISO C90 are disabled) [-std=c90] □

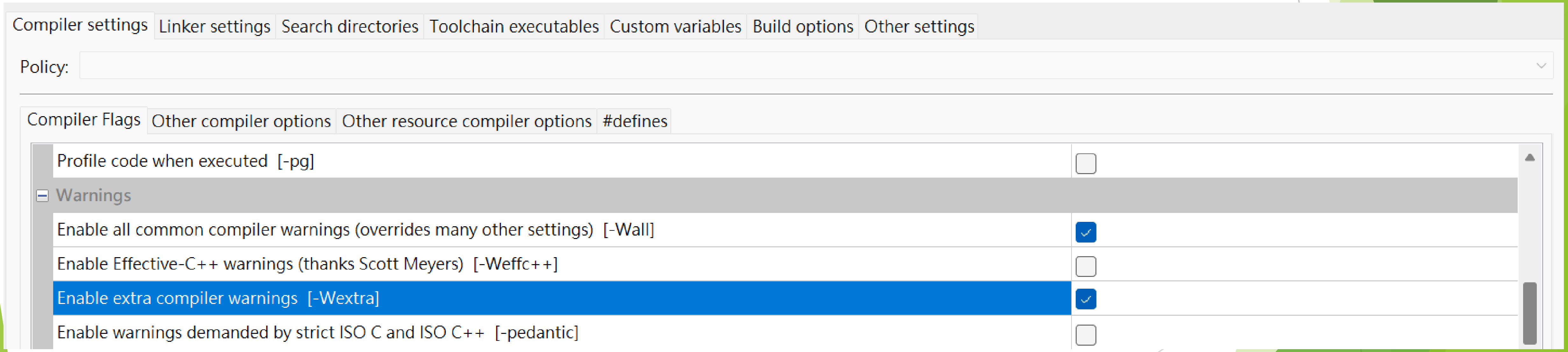
Have gcc follow the 1999 ISO C language standard [-std=c99] □

Have gcc follow the 2011 ISO C language standard [-std=c11] □

Have gcc follow the 2017 ISO C language standard [-std=c17] ☑

Compile

- Slide down the page, you will see the warning section
- Make sure you check the checkbox of –Wall and -Wextra



Profile code when executed [-pg] ☐

Warnings

Enable all common compiler warnings (overrides many other settings) [-Wall] ☑

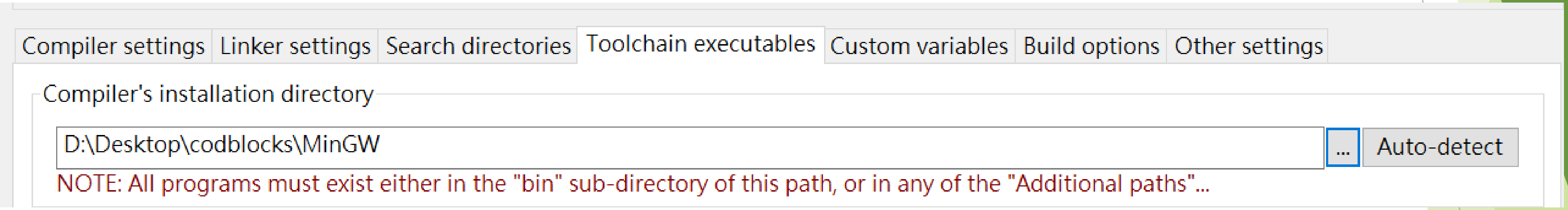
Enable Effective-C++ warnings (thanks Scott Meyers) [-Weffc++] ☐

Enable extra compiler warnings [-Wextra] ☑

Enable warnings demanded by strict ISO C and ISO C++ [-pedantic] ☐

Compile

- In **Toolchain executables** make sure the complier is the MinGW you download with you code blocks



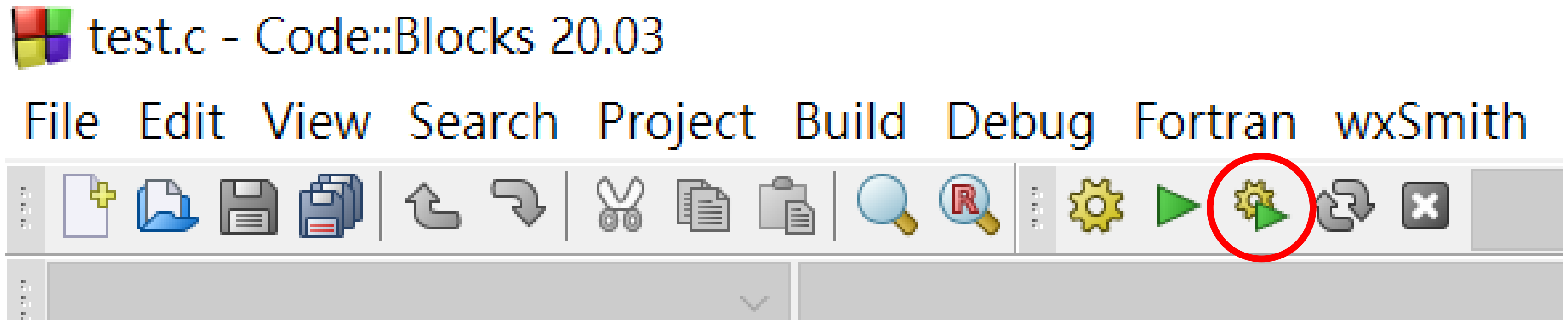
Compiler's installation directory

D:\Desktop\codblocks\MinGW

NOTE: All programs must exist either in the "bin" sub-directory of this path, or in any of the "Additional paths"...

Compile

- Now you can build and run your code by click

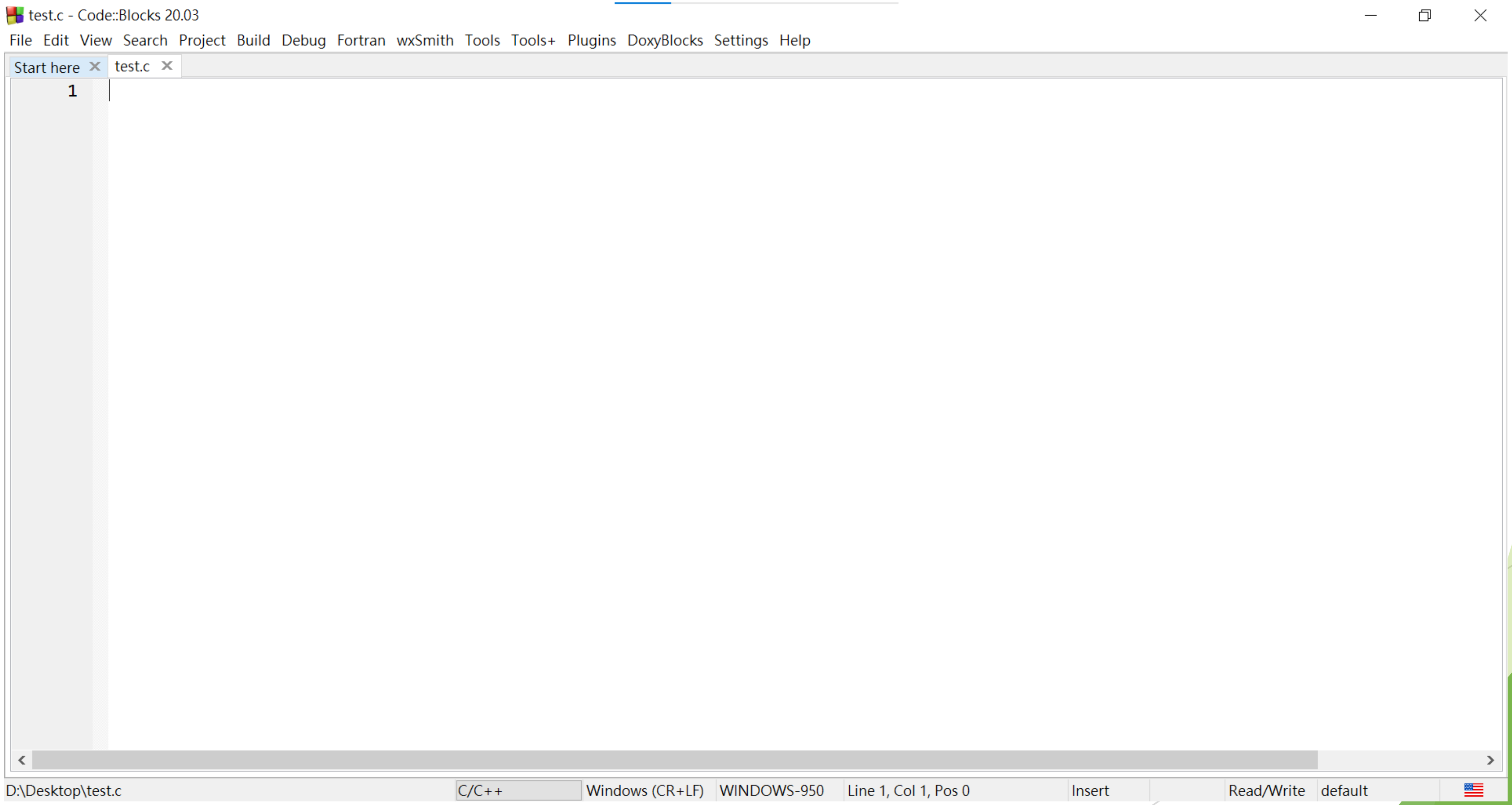


Common problem

- Tool Bar disappear
- Some other tool disappear
- In the management, project tag disappear
- Can’t use Debugger
- Can’t compile the right file
- Multiple main error

Tool Bar disappear

- My tools bar disappear!!

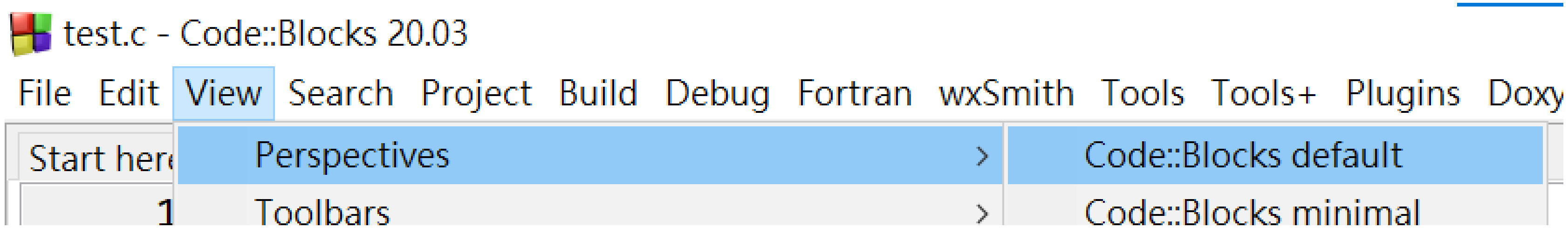


test.c - Code::Blocks 20.03

D:\Desktop\test.c | C/C++ | Windows (CR+LF) | WINDOWS-950 | Line 1, Col 1, Pos 0 | Insert | Read/Write | default

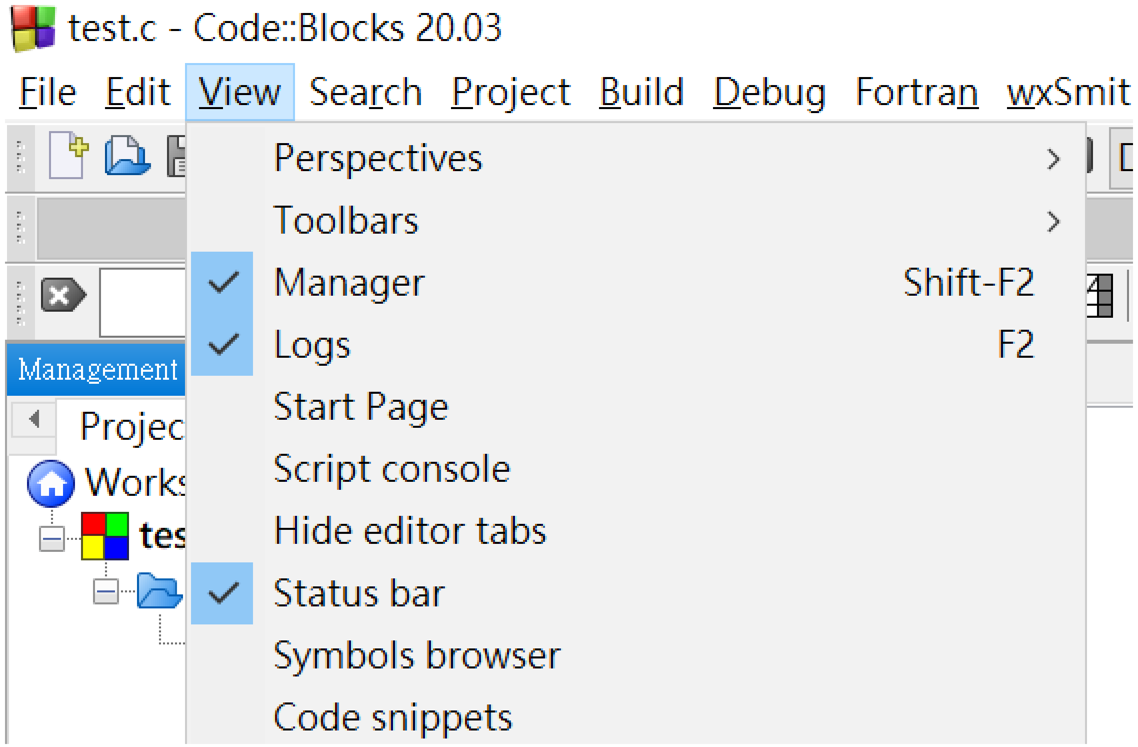
Tool Bar disappear

- Click **View** -> **perspectives** -> **Code::Blocks default**
- If it ask do you want to save, you can either click yes or no

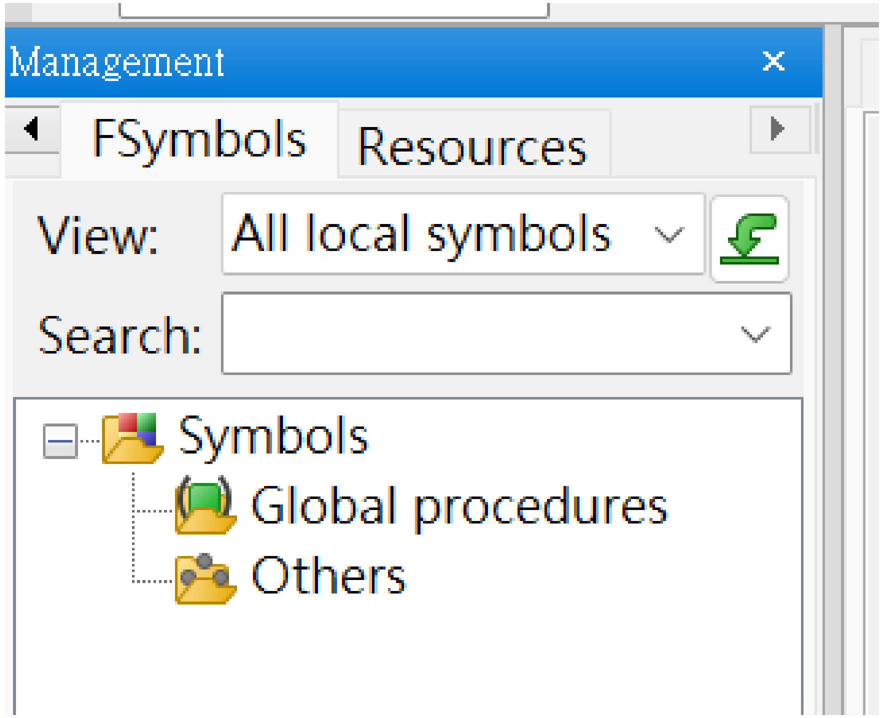

test.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins Doxy
Start here
1
Perspectives
Toolbars
Code::Blocks default
Code::Blocks minimal

Some other tool disappear

- If you can’t find some other tools, check the **View** first

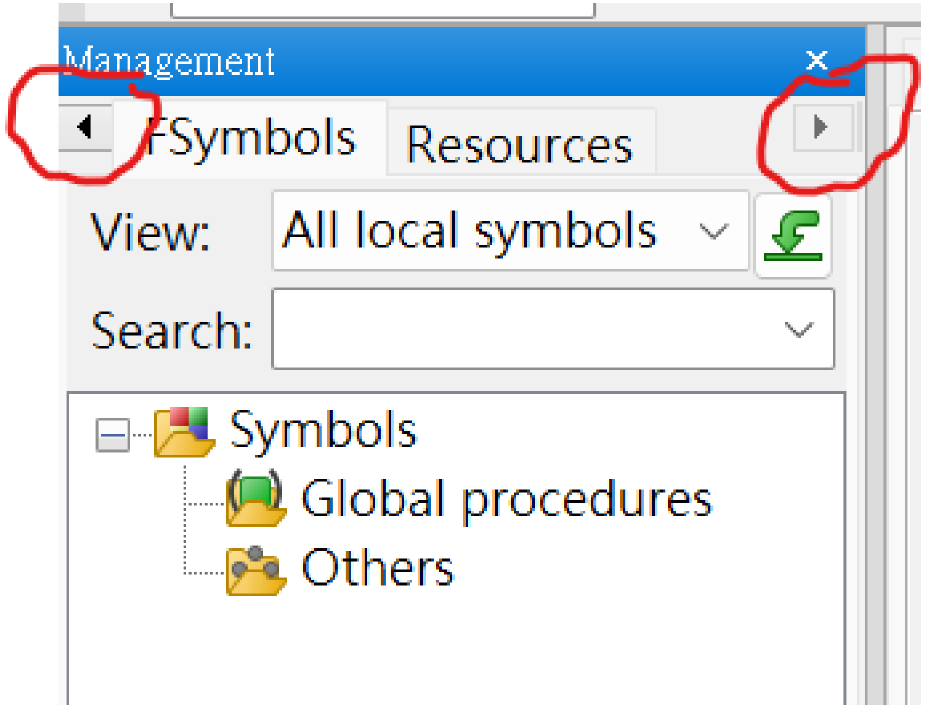


In the management, project tag disappear



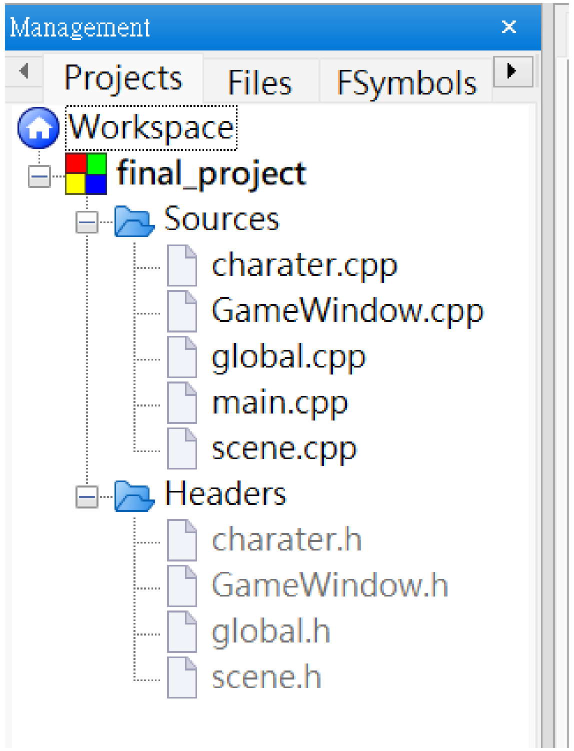
In the management, project tag disappear

- Click these two arrow until you find the project tag

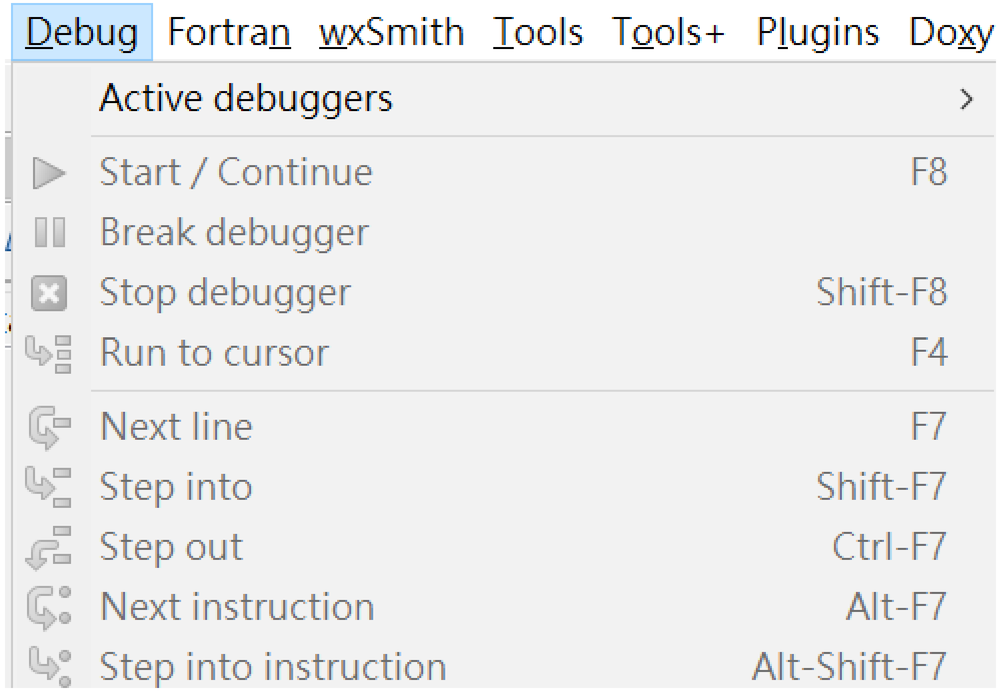


In the management, project tag disappear

- Find the project tag and click the project tag. You can find the interface like below

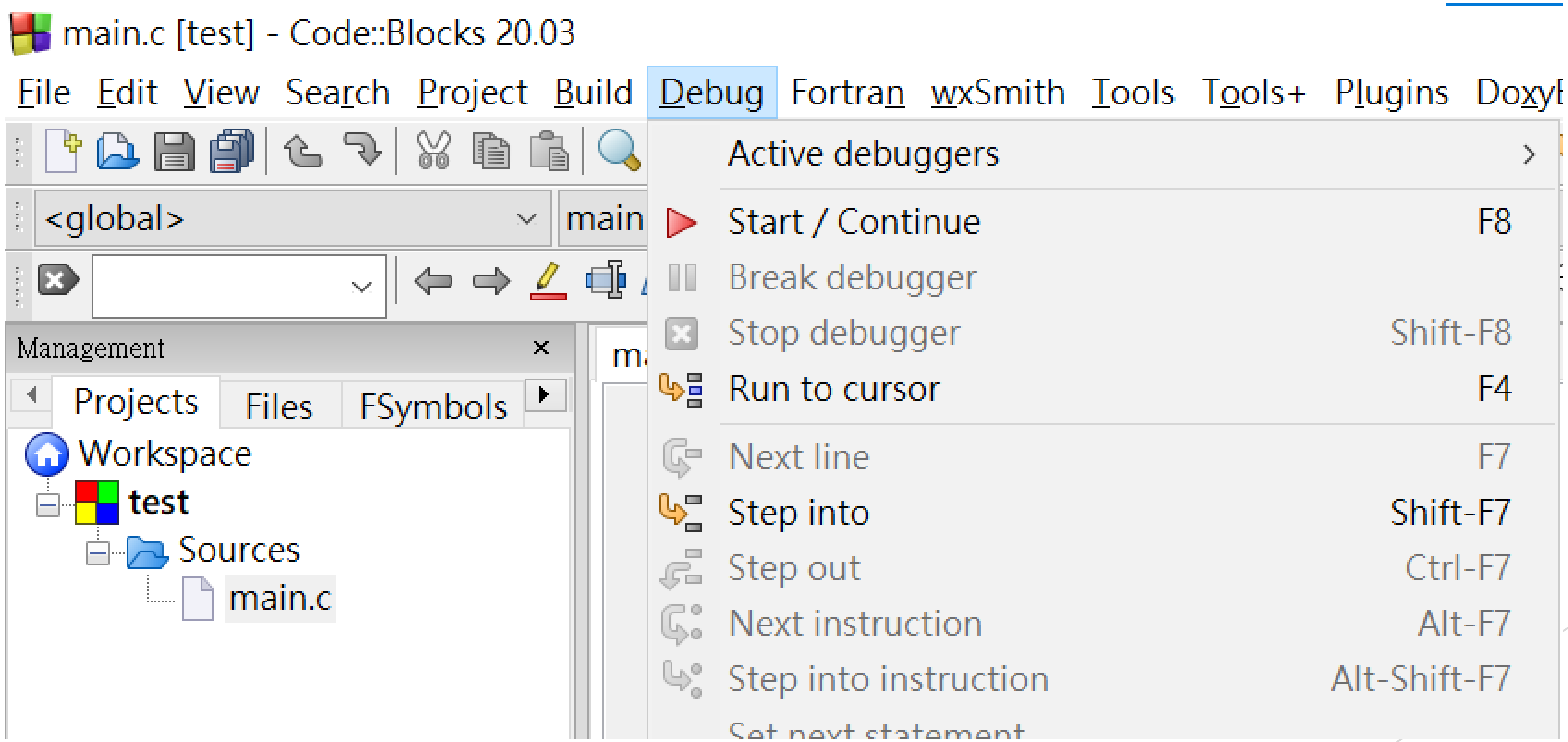


Can’t use Debugger



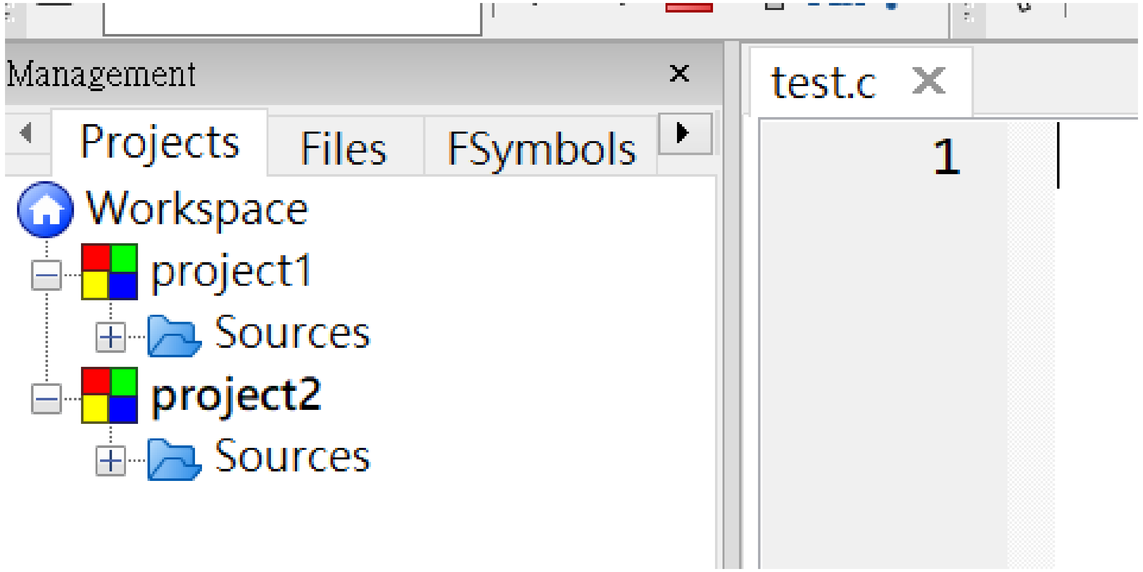
Can’t use Debugger

- It’s because debugger can only be activated in project



Can’t compile the right file

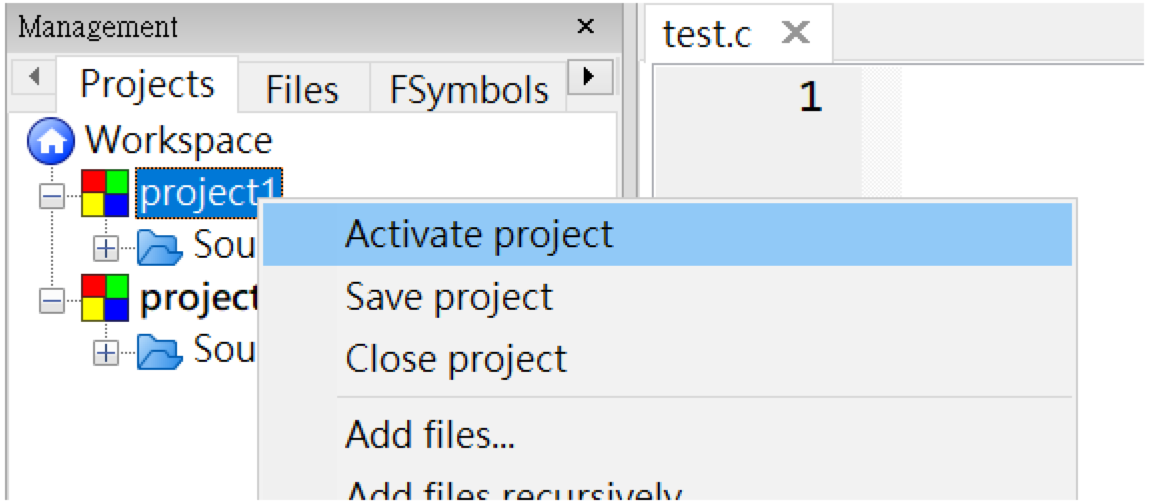
- It may occur when you open your project and some other files simultaneously



- This picture shows 2 project and 1 file test.c(each project has a file **main.c**!!)

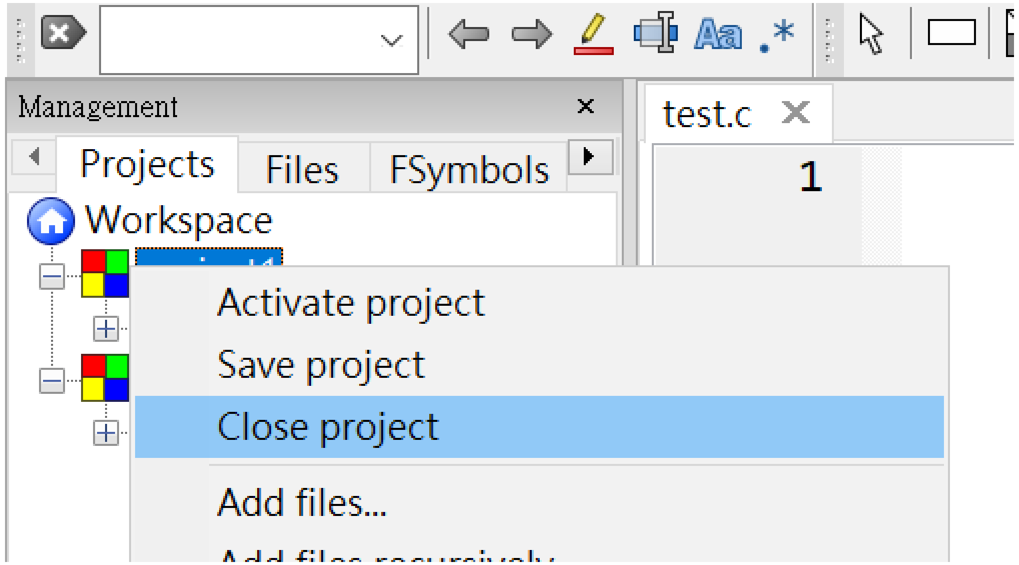
Can’t compile the right file

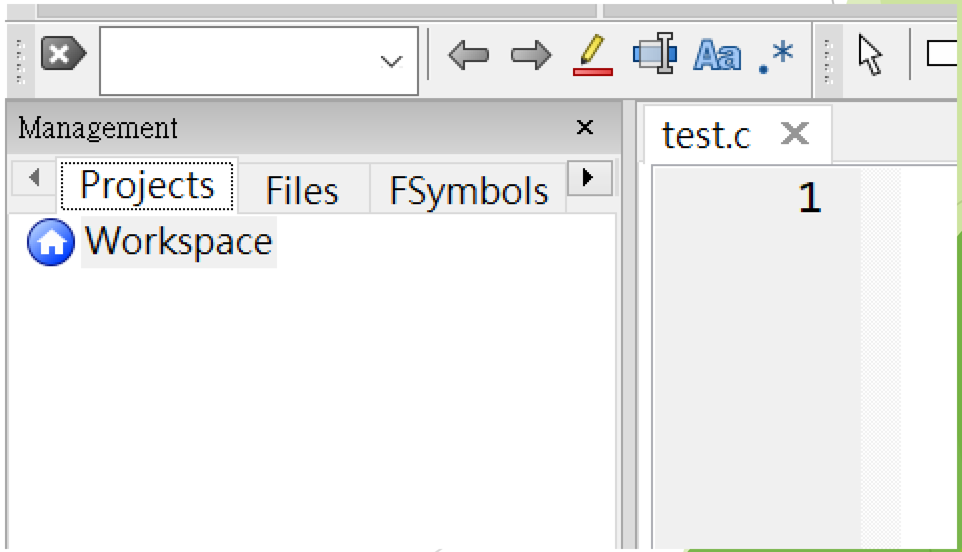
- If you want to compile **project1** right click and activate it



Can’t compile the right file

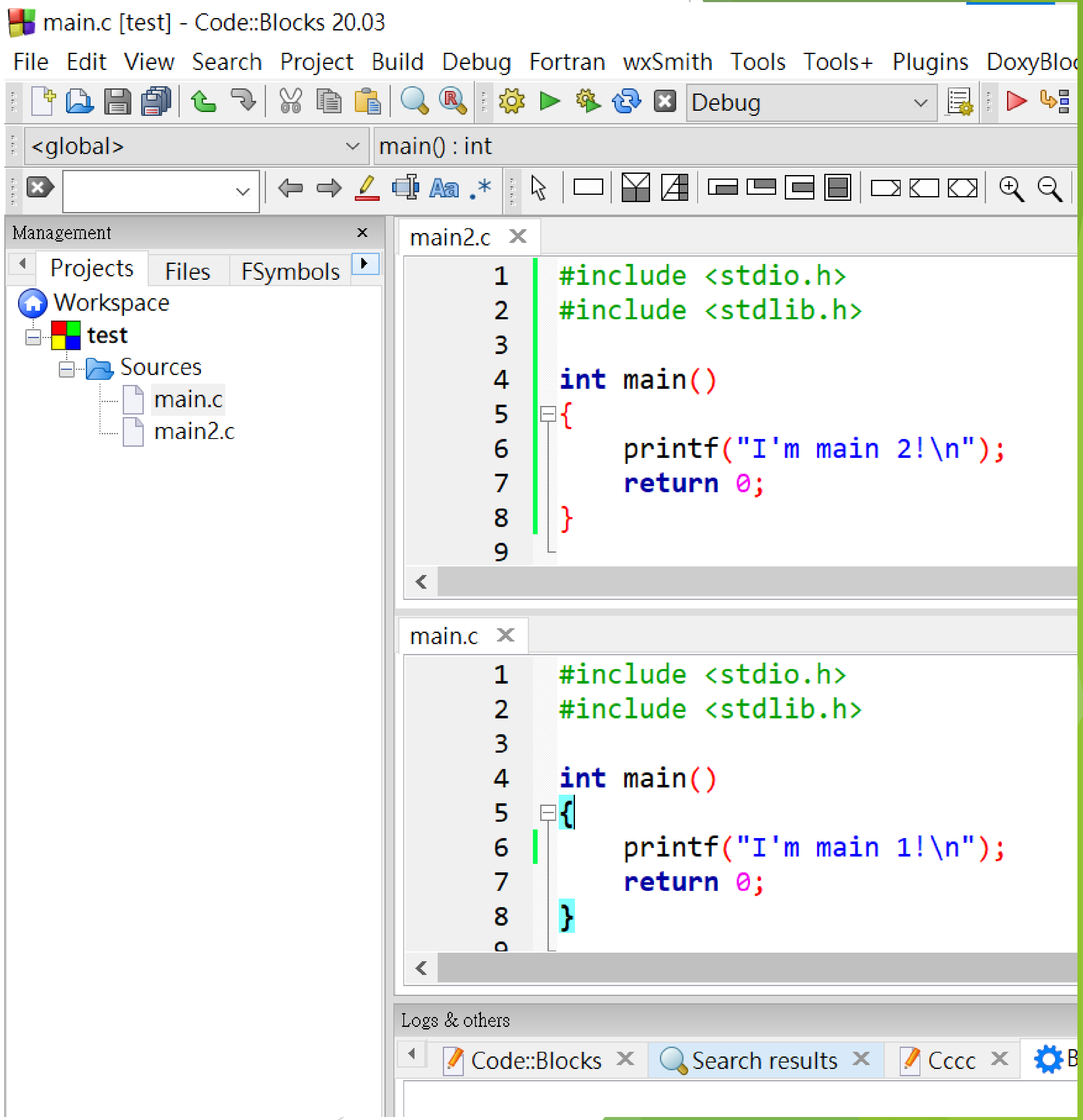
- If you want to compile the **test.c** file, close all project
- Make sure the **Management** -> **Projects** is clear





Multiple main error

- If you have two files in project and both files have **main** function



main2.c

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("I'm main 2!\n");
    return 0;
}
```

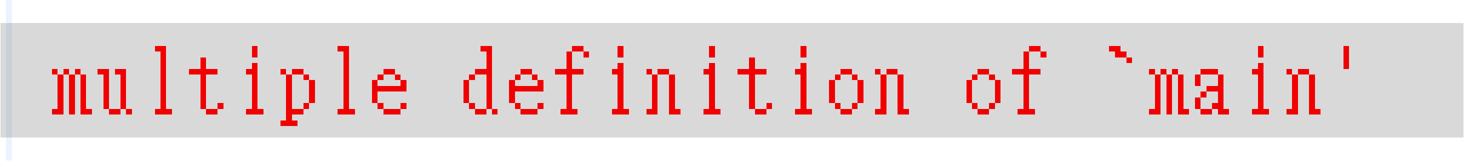
main.c

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("I'm main 1!\n");
    return 0;
}
```

Multiple main error

- A project can only have ONE **main** function



```
multiple definition of `main'
```