



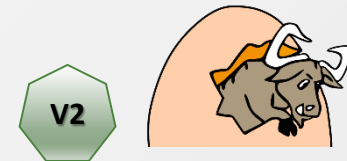
# Ten Minutes to Setup Modern Fortran Compiler 2003/2008

## Modern Fortran Programming

Mohammad Rahmani

Chemical Engineering Department  
Amirkabir University of Technology  
[m.rahmani@aut.ac.ir]

Jan 2016





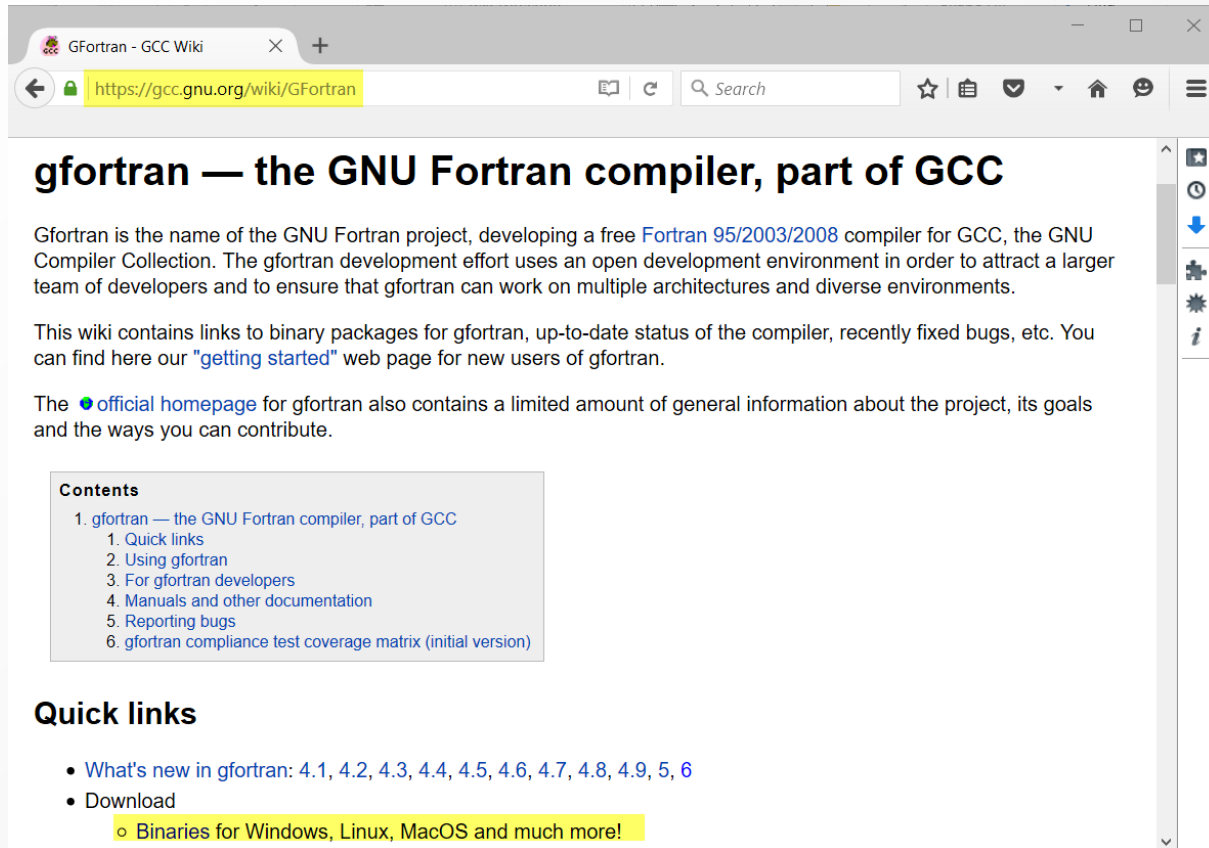
# Outline

- Get the free gfortran compiler 2003/2008
- Get the free IDE, Code::Blocks (CB) for Fortran
- Setup gfortran on Windows
- Setup CB on Windows
- Write your first Fortran program

In this tutorial, we use GFortran as modern Fortran compiler and Code::Blocks as integrated development environment (IDE), both are free and cross platform

# I. Get the gfortran compiler . . .

- Open the link: [gcc.gnu.org/wiki/GFortran](https://gcc.gnu.org/wiki/GFortran)
- Click on **Binaries** for Windows under Download



The screenshot shows a web browser window with the address bar displaying <https://gcc.gnu.org/wiki/GFortran>. The page title is "gfortran — the GNU Fortran compiler, part of GCC". The main content area contains a paragraph about the GNU Fortran project, a paragraph about the wiki's purpose, and a paragraph about the official homepage. Below this is a "Contents" box with a list of links. At the bottom, there is a "Quick links" section with a list of links, including "Binaries for Windows, Linux, MacOS and much more!" which is highlighted in yellow.

**gfortran — the GNU Fortran compiler, part of GCC**

Gfortran is the name of the GNU Fortran project, developing a free [Fortran 95/2003/2008](#) compiler for GCC, the GNU Compiler Collection. The gfortran development effort uses an open development environment in order to attract a larger team of developers and to ensure that gfortran can work on multiple architectures and diverse environments.

This wiki contains links to binary packages for gfortran, up-to-date status of the compiler, recently fixed bugs, etc. You can find here our "[getting started](#)" web page for new users of gfortran.

The [official homepage](#) for gfortran also contains a limited amount of general information about the project, its goals and the ways you can contribute.

**Contents**

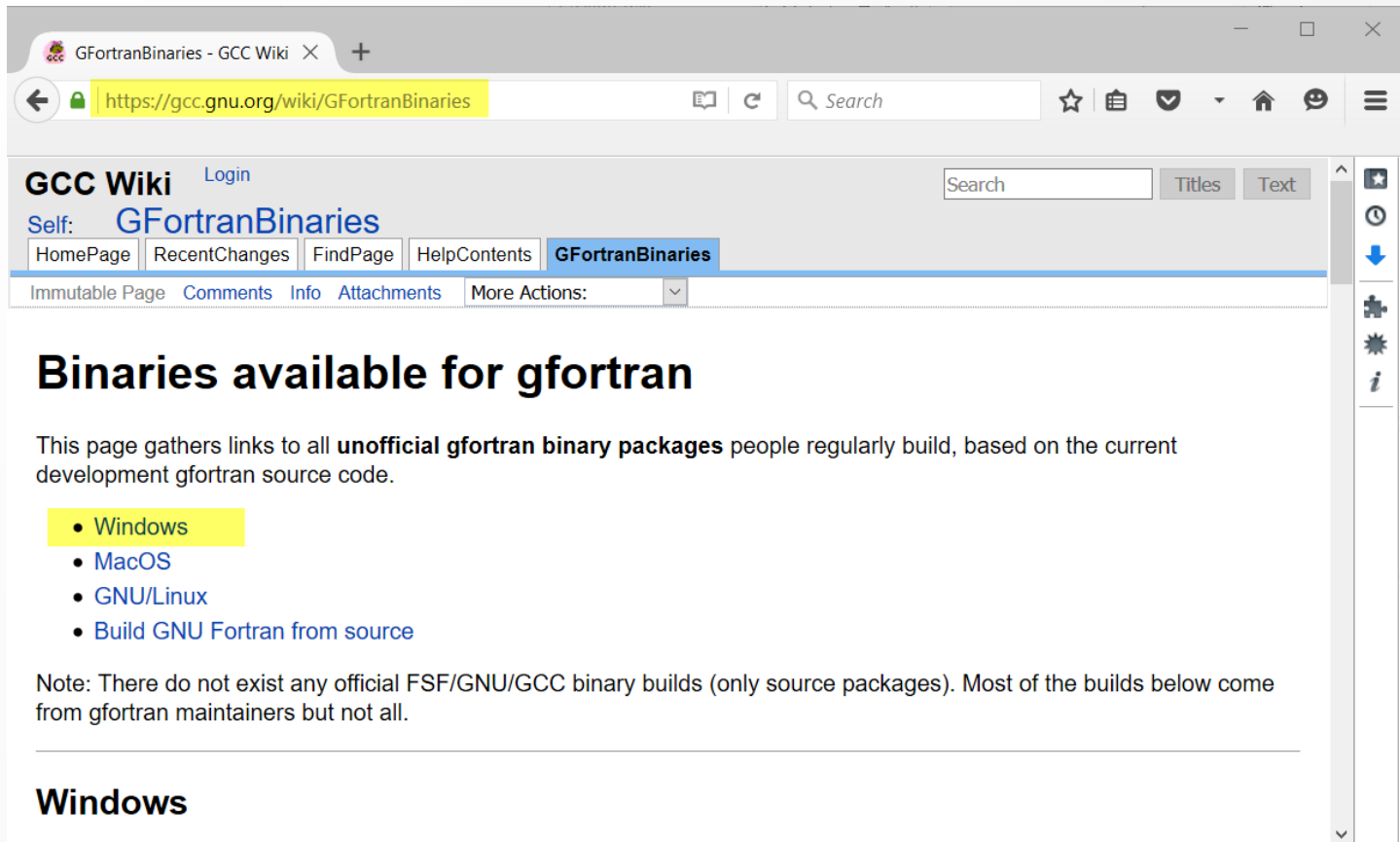
1. [gfortran — the GNU Fortran compiler, part of GCC](#)
  1. [Quick links](#)
  2. [Using gfortran](#)
  3. [For gfortran developers](#)
  4. [Manuals and other documentation](#)
  5. [Reporting bugs](#)
  6. [gfortran compliance test coverage matrix \(initial version\)](#)

**Quick links**

- [What's new in gfortran: 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5, 6](#)
- [Download](#)
  - [Binaries for Windows, Linux, MacOS and much more!](#)

# I. Get the gfortran compiler . . .

- Click on windows:



The screenshot shows a web browser window with the address bar displaying <https://gcc.gnu.org/wiki/GFortranBinaries>. The page title is "GCC Wiki" with a "Login" link. Below the title, there is a search bar and a navigation menu with links: "HomePage", "RecentChanges", "FindPage", "HelpContents", and "GFortranBinaries" (which is highlighted). Underneath the navigation menu, there are links for "Immutable Page", "Comments", "Info", "Attachments", and a "More Actions:" dropdown. The main content area has the heading "Binaries available for gfortran". Below this heading, a paragraph states: "This page gathers links to all **unofficial gfortran binary packages** people regularly build, based on the current development gfortran source code." A bulleted list follows, with the first item "Windows" highlighted in yellow. The other items are "MacOS", "GNU/Linux", and "Build GNU Fortran from source". A note at the bottom of the list says: "Note: There do not exist any official FSF/GNU/GCC binary builds (only source packages). Most of the builds below come from gfortran maintainers but not all." Below the note, the heading "Windows" is visible.

GCC Wiki Login

Self: [GFortranBinaries](#)

[HomePage](#) [RecentChanges](#) [FindPage](#) [HelpContents](#) [GFortranBinaries](#)

[Immutable Page](#) [Comments](#) [Info](#) [Attachments](#) [More Actions:](#)

## Binaries available for gfortran

This page gathers links to all **unofficial gfortran binary packages** people regularly build, based on the current development gfortran source code.

- Windows
- MacOS
- GNU/Linux
- Build GNU Fortran from source

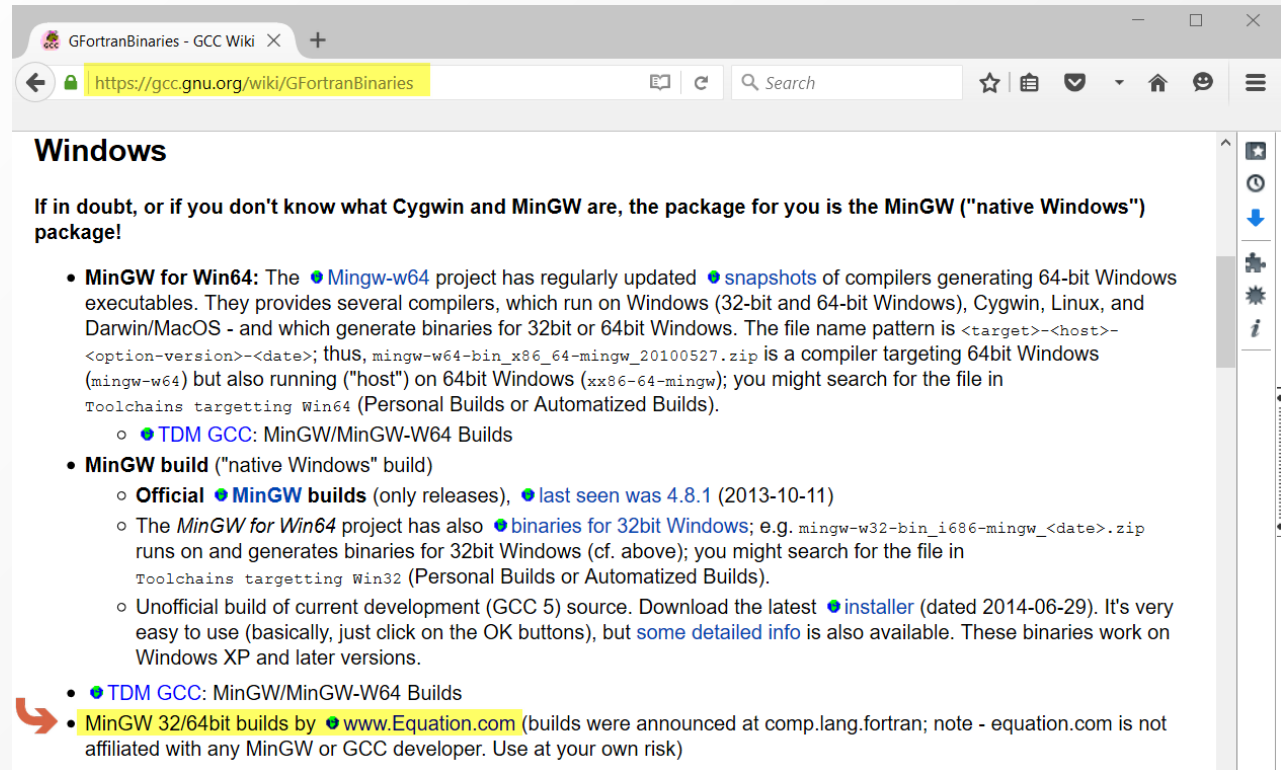
Note: There do not exist any official FSF/GNU/GCC binary builds (only source packages). Most of the builds below come from gfortran maintainers but not all.

### Windows

# I. Get the gfortran compiler . . .

- There are different binaries by different providers to download:
  - 32 bit
  - 64 bit

For this demo, we download the latest [MinGW](#) 64 bit version from [equation.com](#) to work on Windows 10!



The screenshot shows a web browser window with the address bar displaying <https://gcc.gnu.org/wiki/GFortranBinaries>. The page title is "GFortranBinaries - GCC Wiki". The main heading is "Windows". Below the heading, the text reads: "If in doubt, or if you don't know what Cygwin and MinGW are, the package for you is the MinGW ("native Windows") package!". The page contains several bullet points and links:

- **MinGW for Win64:** The [Mingw-w64](#) project has regularly updated [snapshots](#) of compilers generating 64-bit Windows executables. They provide several compilers, which run on Windows (32-bit and 64-bit Windows), Cygwin, Linux, and Darwin/MacOS - and which generate binaries for 32bit or 64bit Windows. The file name pattern is `<target>-<host>-<option-version>-<date>`; thus, `mingw-w64-bin_x86_64-mingw_20100527.zip` is a compiler targeting 64bit Windows (mingw-w64) but also running ("host") on 64bit Windows (xx86-64-mingw); you might search for the file in Toolchains targeting Win64 (Personal Builds or Automatized Builds).
  - [TDM GCC](#): MinGW/MinGW-W64 Builds
- **MinGW build ("native Windows" build)**
  - **Official** [MinGW builds](#) (only releases), [last seen was 4.8.1](#) (2013-10-11)
  - The [MinGW for Win64](#) project has also [binaries for 32bit Windows](#); e.g. `mingw-w32-bin_i686-mingw_<date>.zip` runs on and generates binaries for 32bit Windows (cf. above); you might search for the file in Toolchains targeting Win32 (Personal Builds or Automatized Builds).
  - Unofficial build of current development (GCC 5) source. Download the latest [installer](#) (dated 2014-06-29). It's very easy to use (basically, just click on the OK buttons), but [some detailed info](#) is also available. These binaries work on Windows XP and later versions.
- [TDM GCC](#): MinGW/MinGW-W64 Builds
- [MinGW 32/64bit builds by \[www.Equation.com\]\(#\)](#) (builds were announced at [comp.lang.fortran](#); note - equation.com is not affiliated with any MinGW or GCC developer. Use at your own risk)

*You can later download other builds and follow the same instructions*



## II. Get the IDE

- If you don't need an IDE and use simple editor to code! you can skip this step.
- There are two different IDEs you can download for free:
  - Code::Blocks (CB) for Fortran
    - Easy to use, syntax highlighting and code completion
    - Support multi-projects workspace
    - Support building static and dynamic libraries
    - Support debugging
  - Photran
    - A great portable IDE based on Eclipse (need JRE)
    - Refactoring capabilities
    - Support debugging, workspace
    - Support building static and dynamic libraries
- Here, CB is selected as the Fortran IDE

*IDE stands for Integrated Development Environment*

## II. Get the CB IDE for Fortran ...

- Open the CB link: <http://codeblocks.org/>
- Click on download



## II. Get the CB IDE for Fortran ...

- Download the appropriate build

we have put in the new release.

**Windows 2000 / XP / Vista / 7:**

File	Date	Download from
codeblocks-13.12-setup.exe <b>1</b>	27 Dec 2013	BerliOS or Sourceforge.net
codeblocks-13.12mingw-setup.exe <b>2</b>	27 Dec 2013	BerliOS or Sourceforge.net
codeblocks-13.12mingw-setup-TDM-GCC-481.exe	27 Dec 2013	BerliOS or Sourceforge.net

**NOTE:** The codeblocks-13.12mingw-setup.exe file *includes* the GCC compiler and GDB debugger from **TDM-GCC** (version 4.7.1, 32 bit). The codeblocks-13.12mingw-setup-TDM-GCC-481.exe file includes the TDM-GCC compiler, version 4.8.1, 32 bit. While v4.7.1 is rock-solid (we use it to compile C::B), v4.8.1 is provided for convenience, there are some known bugs with this version related to the compilation of Code::Blocks itself.

IF UNSURE, USE "codeblocks-13.12mingw-setup.exe"

**Linux 32-bit:**

1. Here, we download CB 15.12 (nightly version), for Windows. CB is also available for other platforms

2. You can also download CB with gcc (MinGW) included!

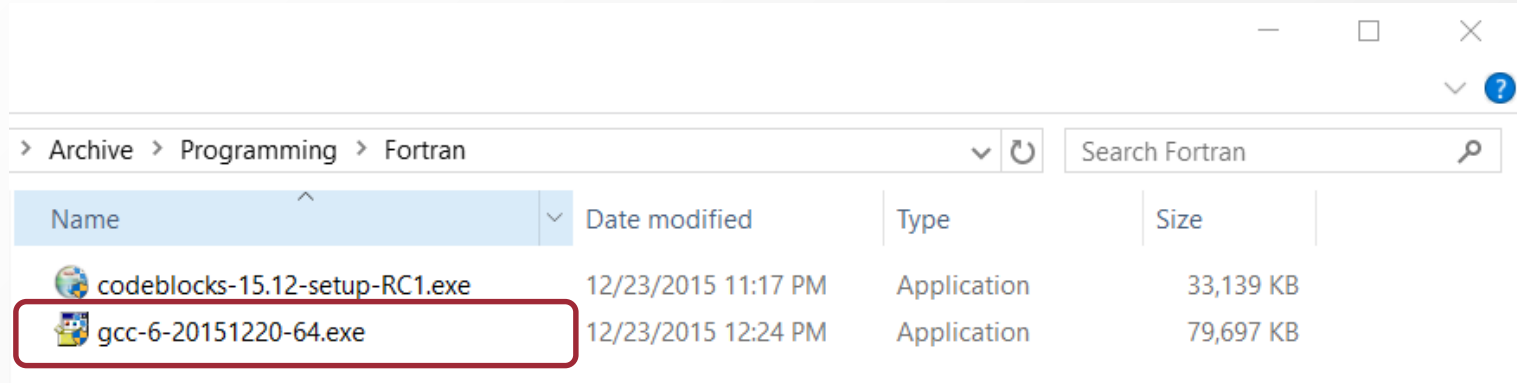
3. The latest nightly version can be downloaded from Nightlies!



### III. Setup gfortran ...

- First install gfortran on Windows by running the executable downloaded in step I

*The file version on the date of this presentation is gcc-6-20151220-64*



*In this tutorial, we install gcc on Windows 10, 64 bit. The instruction is the same under other versions of Windows!*

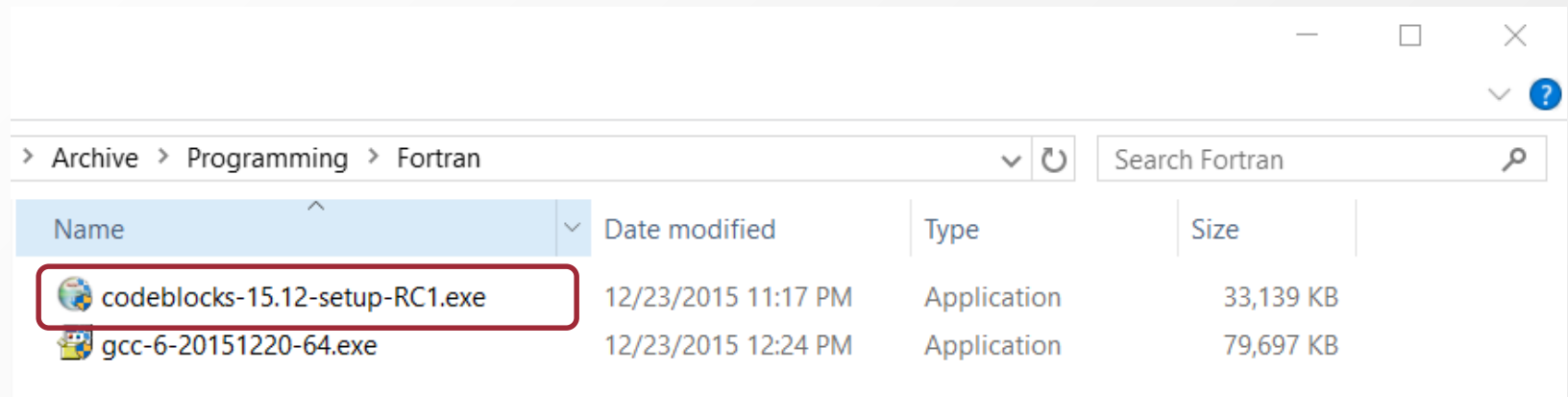


### III. Setup gfortran ...

- Follow the on screen instruction to complete the install!
- By default gfortran will be installed in Windows “Program Files” folder. If you setup a 32 bit version on 64 bit Windows, it will be installed in “Program Files (x86)”
- I recommend to choose an installation folder separate from Windows Program Files! But it works fine even you install in “Program Files” folder!
- For this tutorial, we have installed the gcc compiler collection in *D:\gfortran\gcc6*

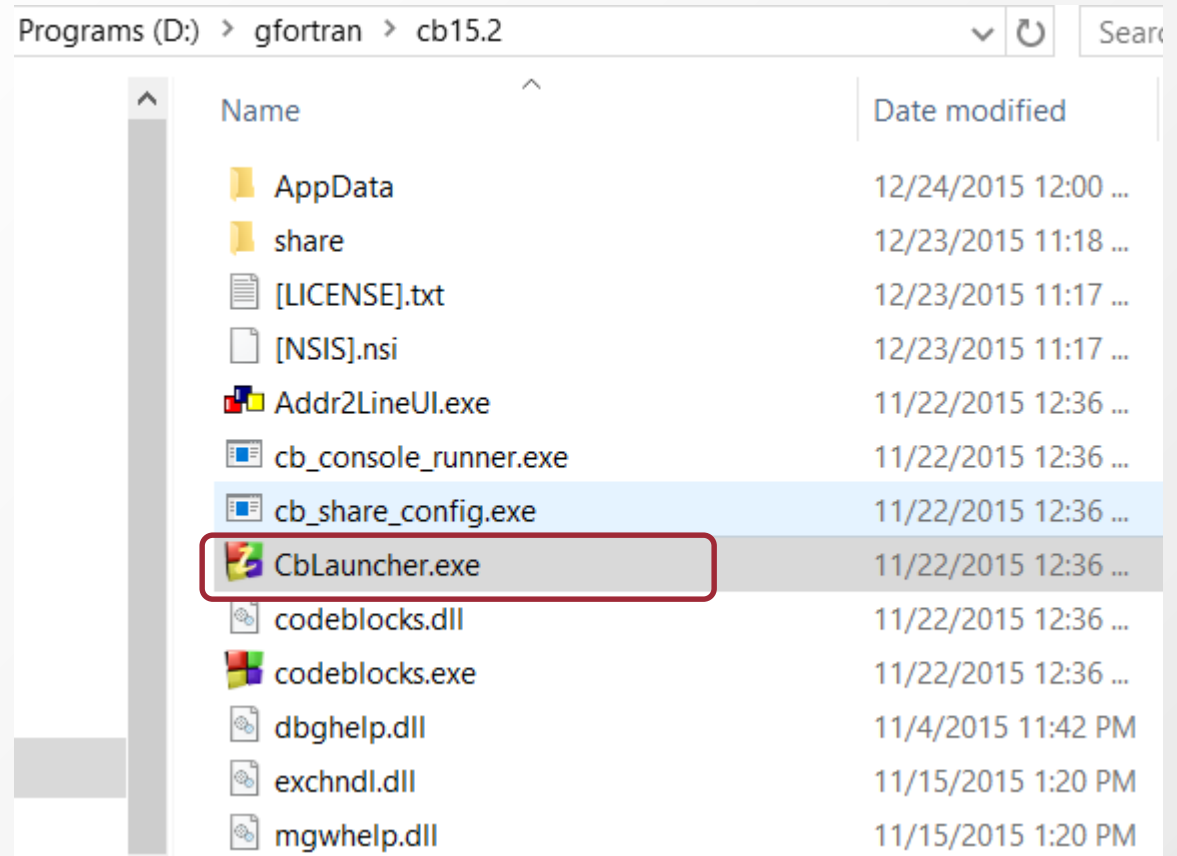
## IV. Setup CB for Fortran ...

- Next, install Code::Blocks (CB)
- For this tutorial we used the latest release (15.12 RC1)
- *CB and gcc both are portable and can be installed by simple unzip to a folder of choice. (you can use the free 7-zip to do this).*



# Run CB and Set Options

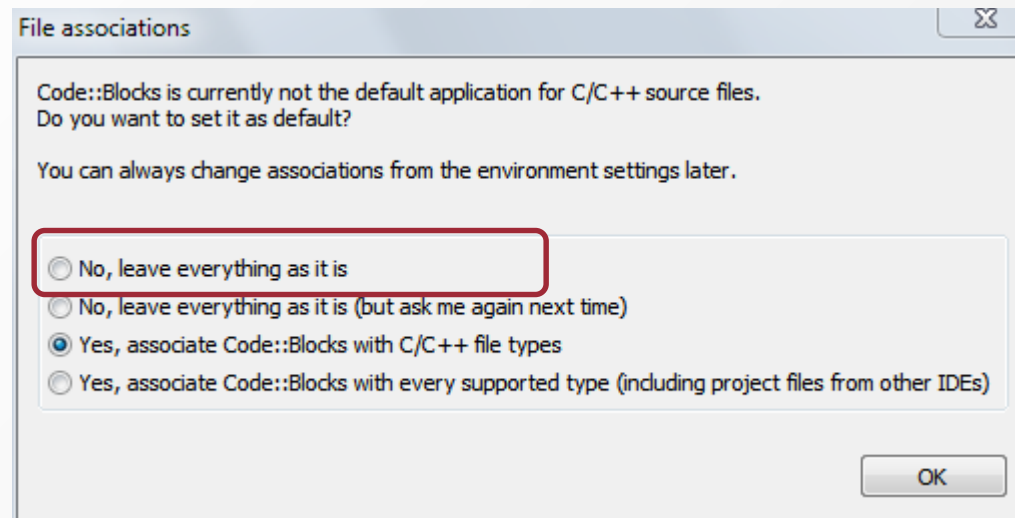
- Open the CB installation folder and run codeblocks.exe
- For convenience, you can make a shortcut on your desktop!



*You can use CB as portable. To run CB in portable mode run "CbLauncher.exe"*

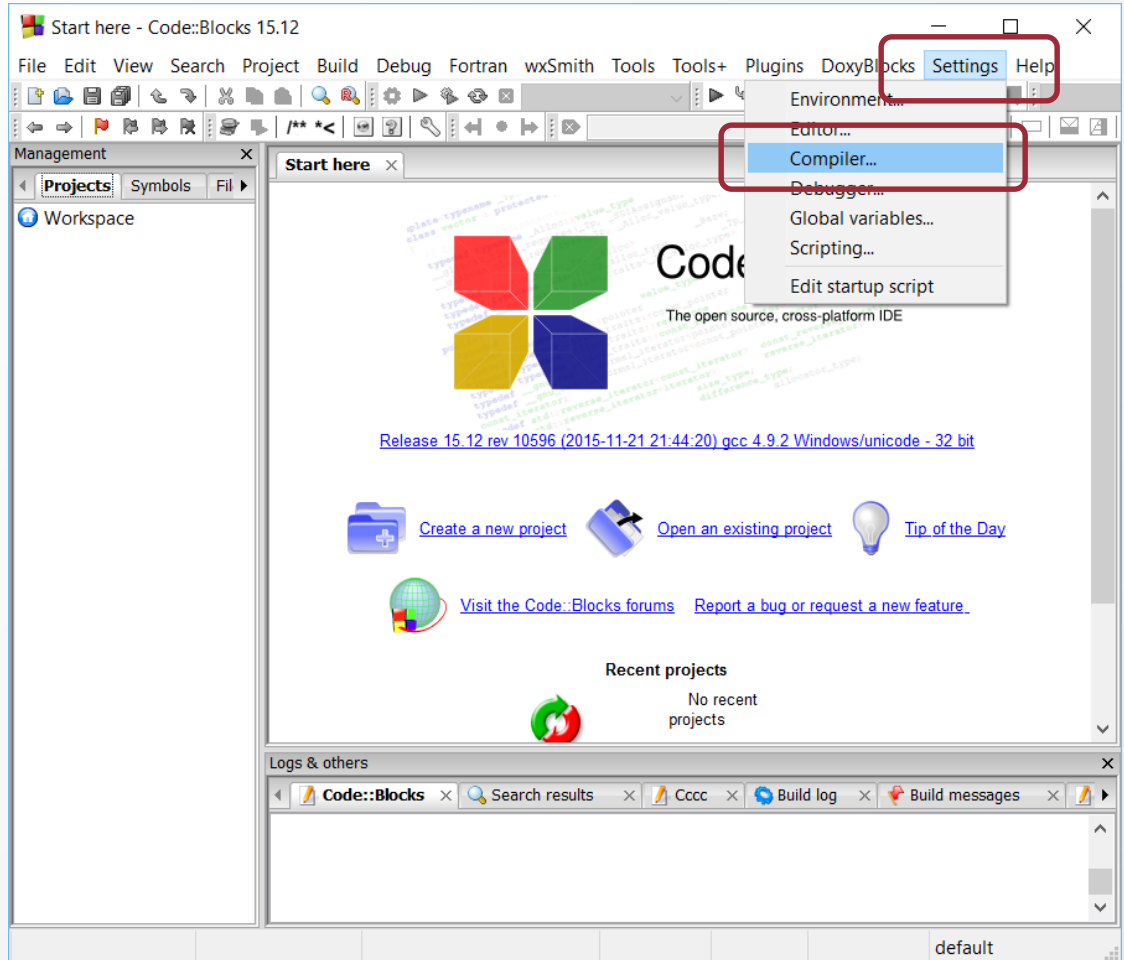
# File associations

- CB lets you to associates file types with itself.
- You can skip this step or set some file types to be opened by CB.



# Set compiler path ...

- In CB, click on setting
- Select 'Compiler'

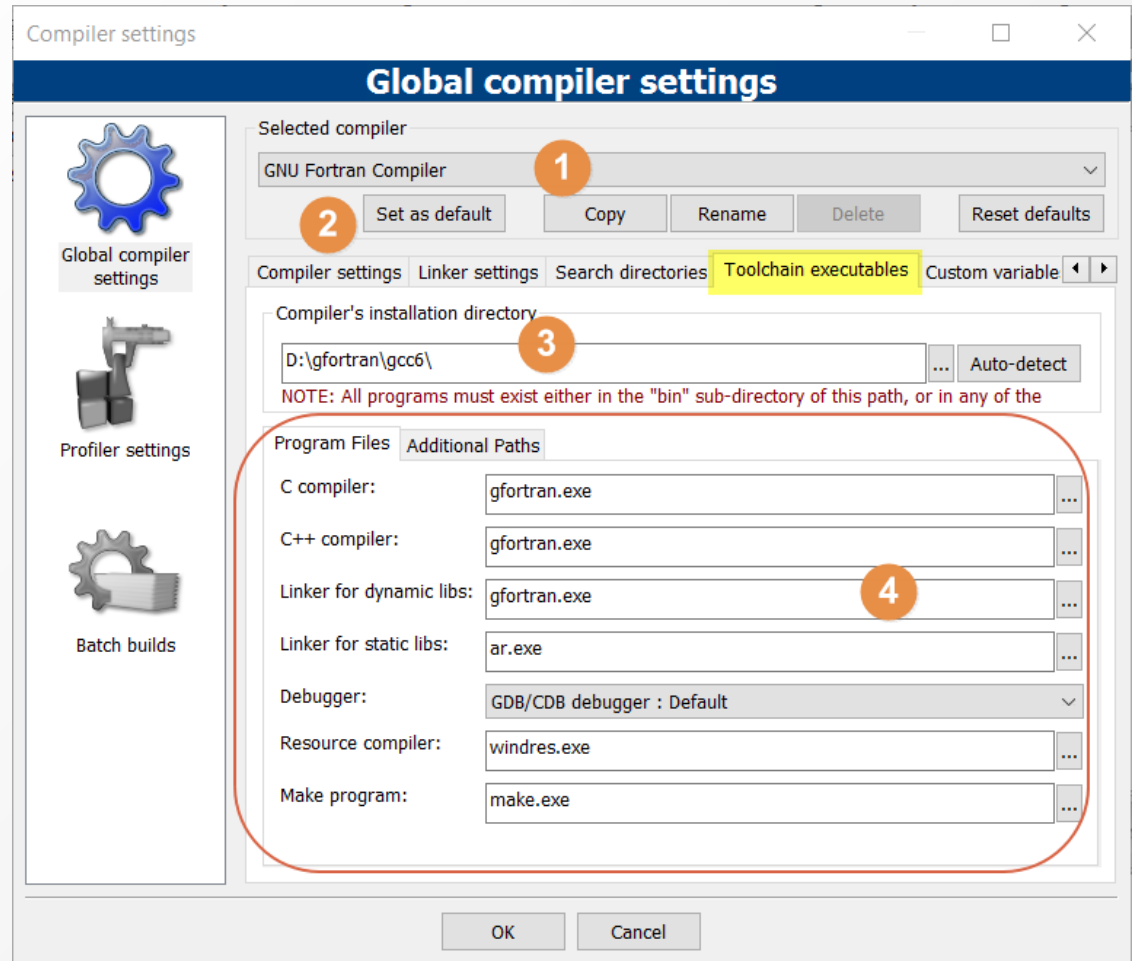


# Set compiler path

- From selected compiler, choose “GNU Fortran Compiler”
- Click on  
“Set as default”

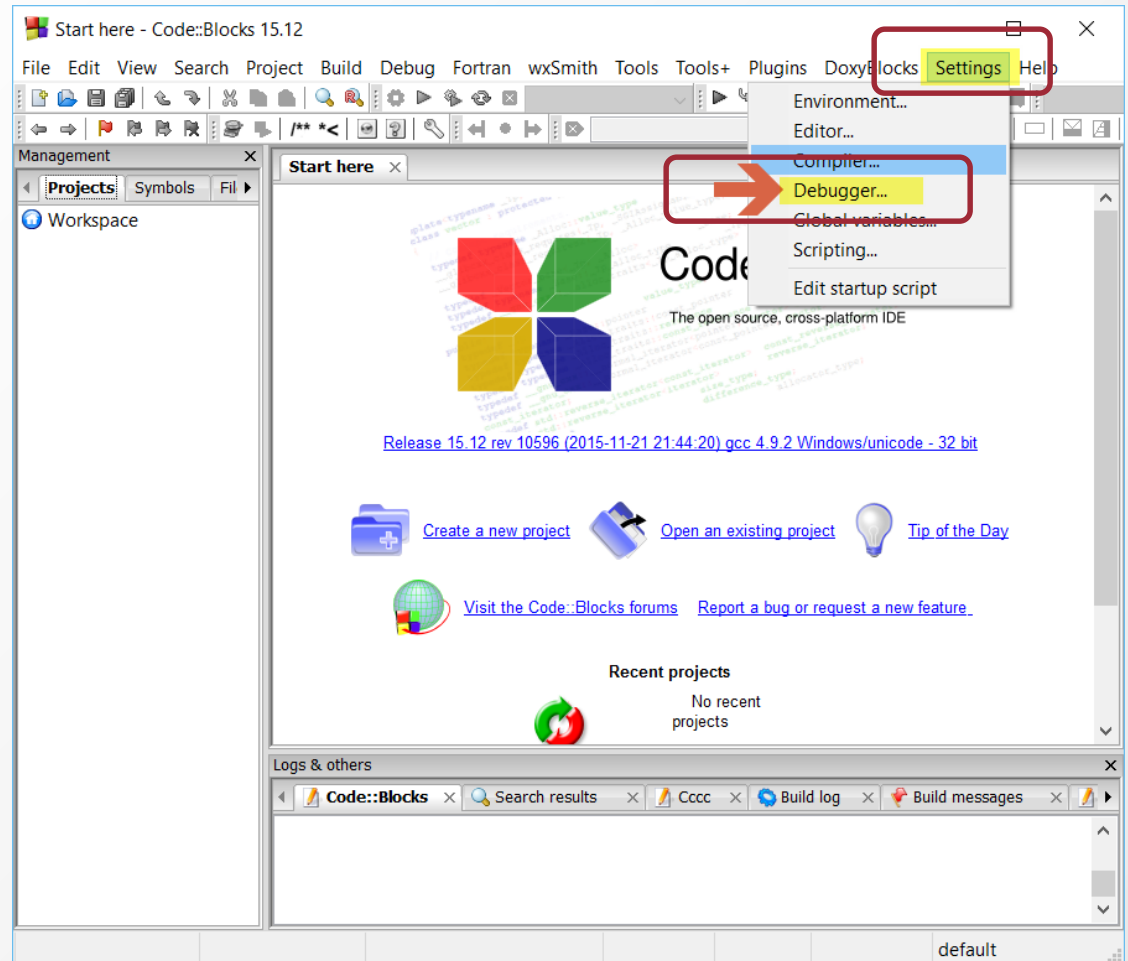
1. Click on ‘Toolchain executables’ and set the path  
2. For this demo, the path is as D:\gfortran\gcc (3)!  
Correct it based on your installation  
3. Correct the program files as shown in step 4 in figure.

Click, OK, that’s all



# Set debugger path ...

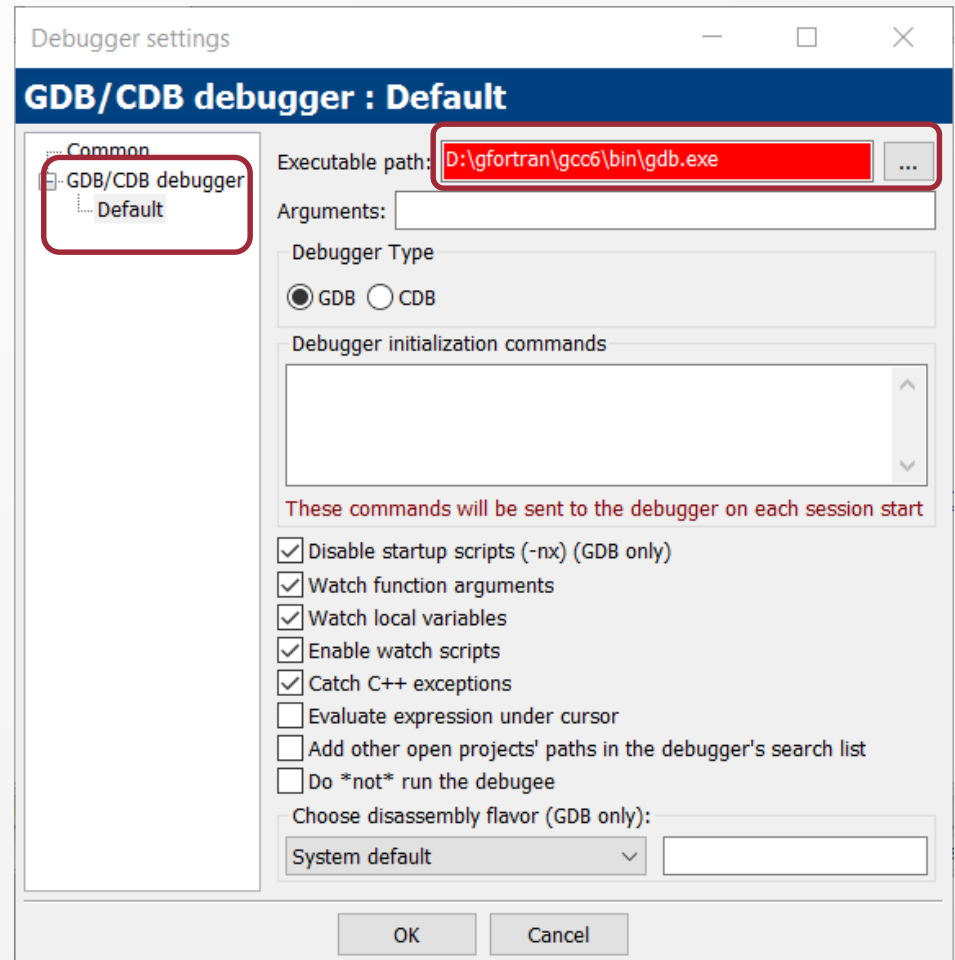
- In CB, click on setting
- Select 'Debugger'





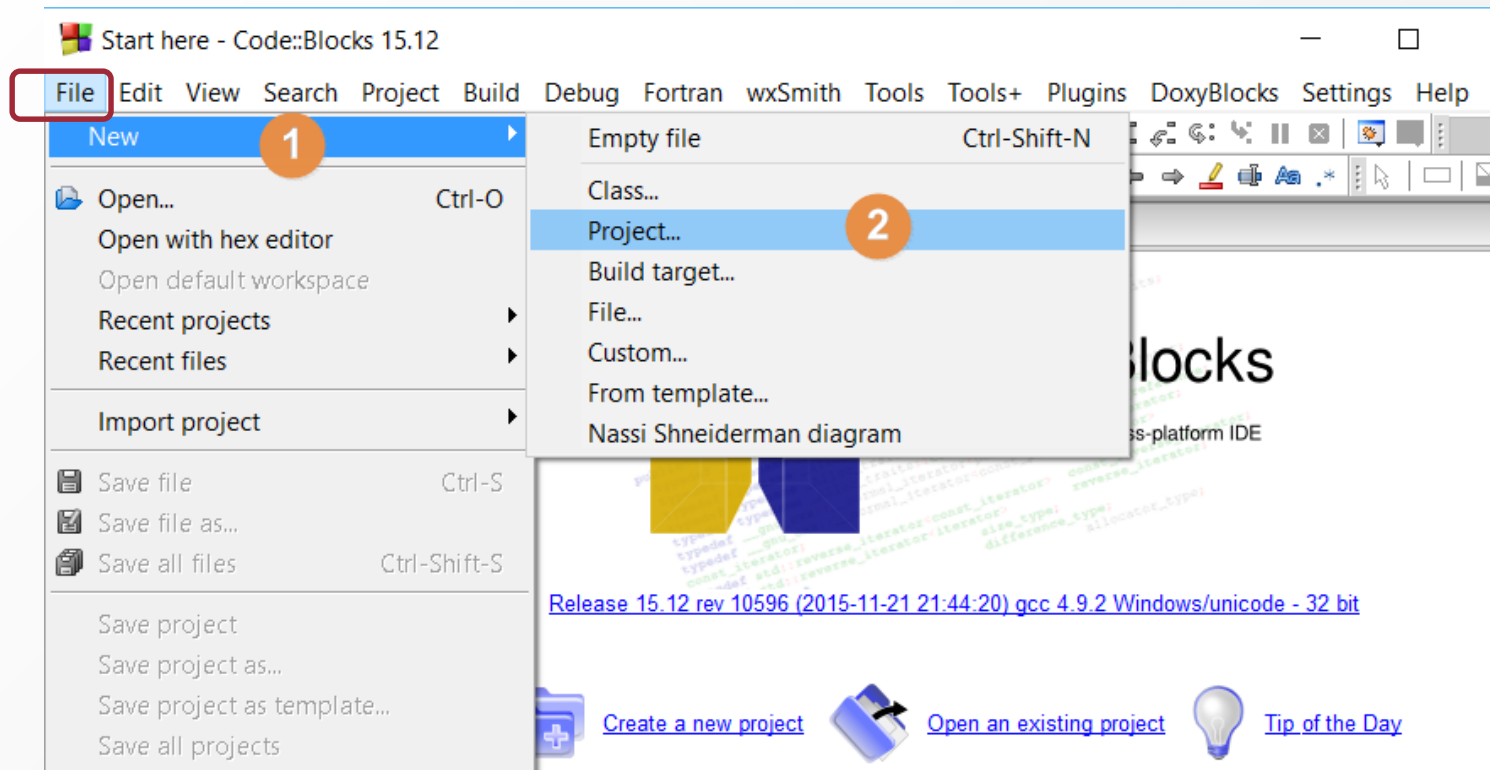
# Set debugger path ...

- Set the Executable path for debugger (gdb.exe)
- In this demo our path is as `D:\gfortran\gcc6\bin\gdb.exe`
- Click OK



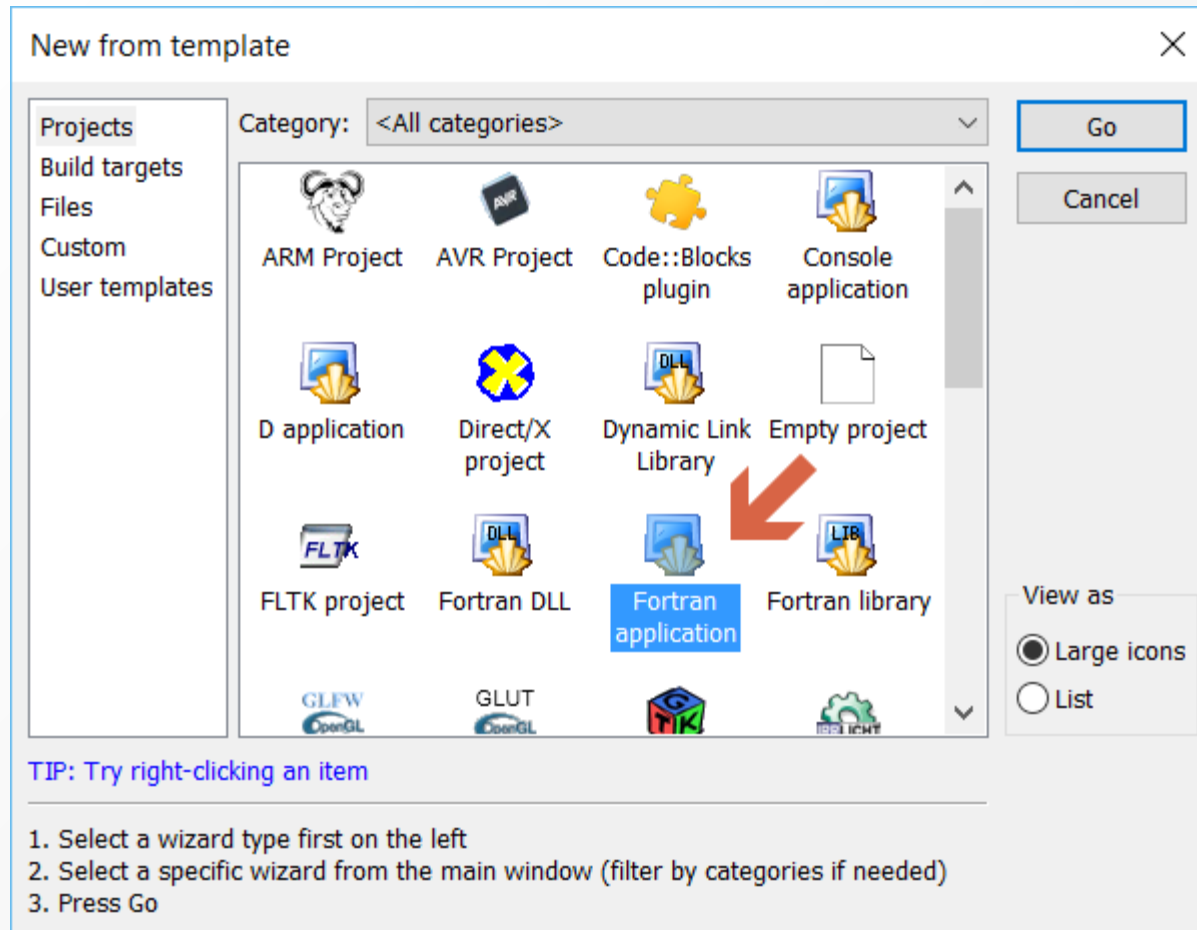
## V. Create a simple Fortran program ...

- Click on 'New' from File on Menu bar
- Select 'Project'



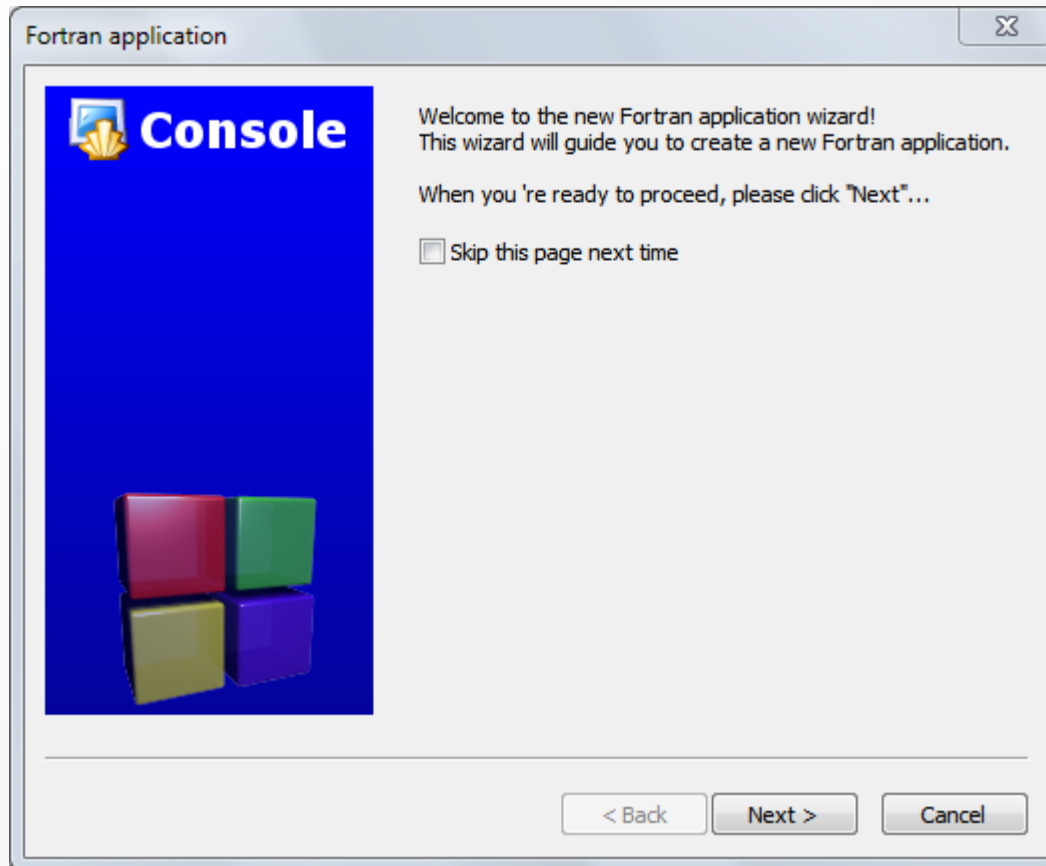
# Fortran Application

- In the window appears, select 'Fortran Application'



# Welcome window

- Select 'Next'



# Select project title and folder

- Enter a project title (1)
- Select a project folder (2)
- Click next (3)

Fortran application

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

Project title: HelloWorld 1

Folder to create project in: D:\gfortran\work\ 2

Project filename: HelloWorld.cbp

Resulting filename: D:\gfortran\work\HelloWorld\HelloWorld.cbp

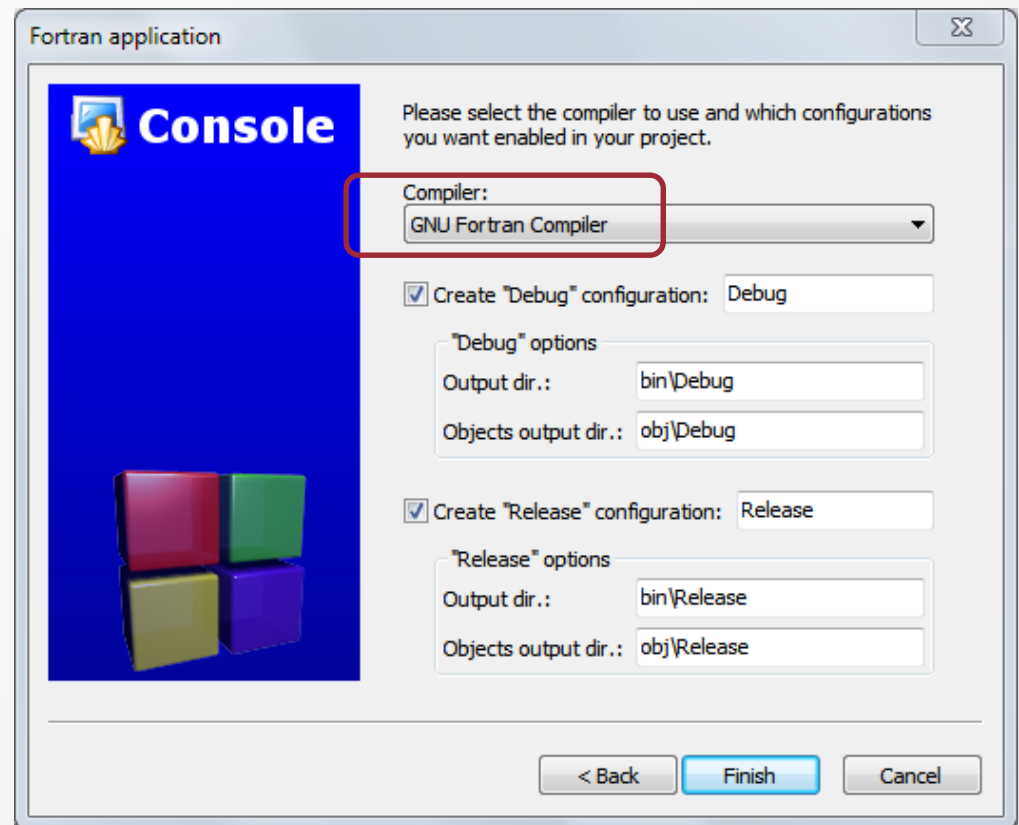
3

< Back Next > Cancel

*Don't worry about the Project filename, it will be 'main.f90'*

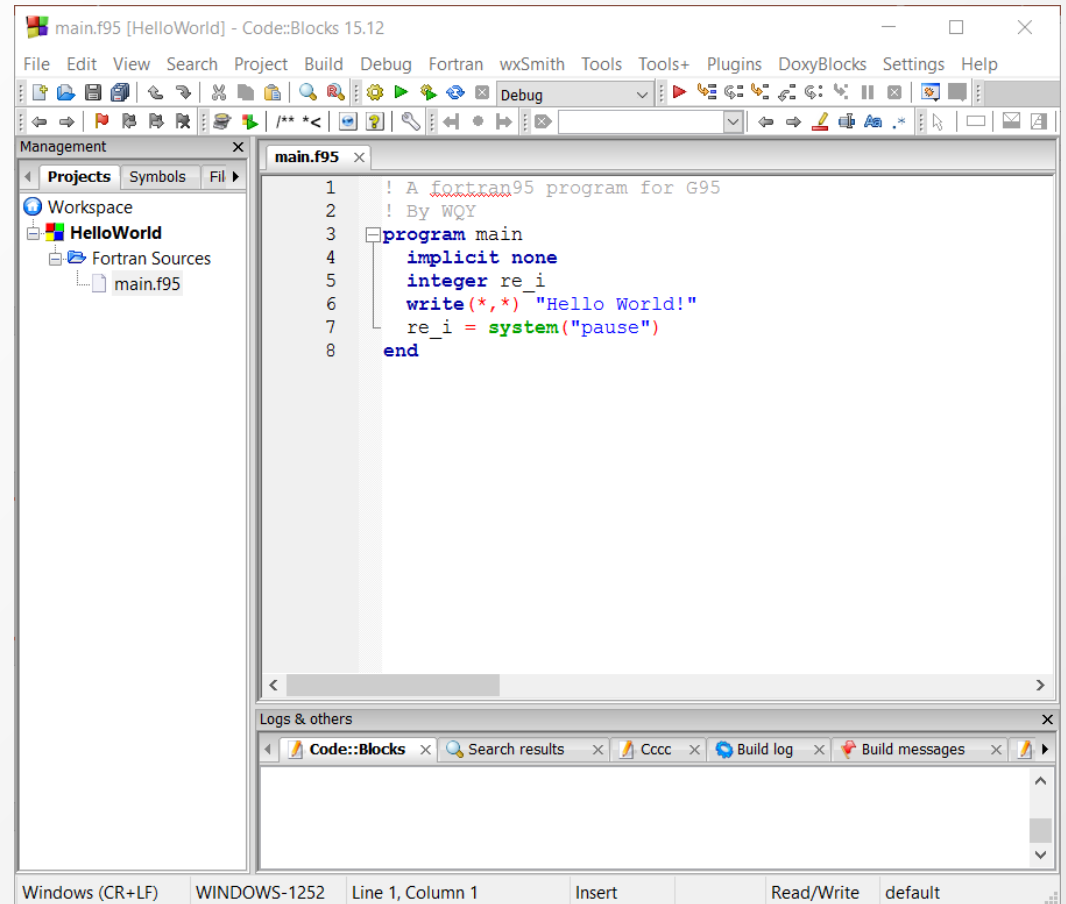
# Fortran project configuration

- By default CB create two build configurations
  - Debug configuration
  - Release configuration
- Make sure 'Compiler' is set to *'GNU Fortran Compiler'*
- Click on 'Finish'



# The Fortran source file

- By default a 'main.f95' is created
- You can edit it and put your own code inside 'main.f95'
- You can also rename it, if you like
- ...



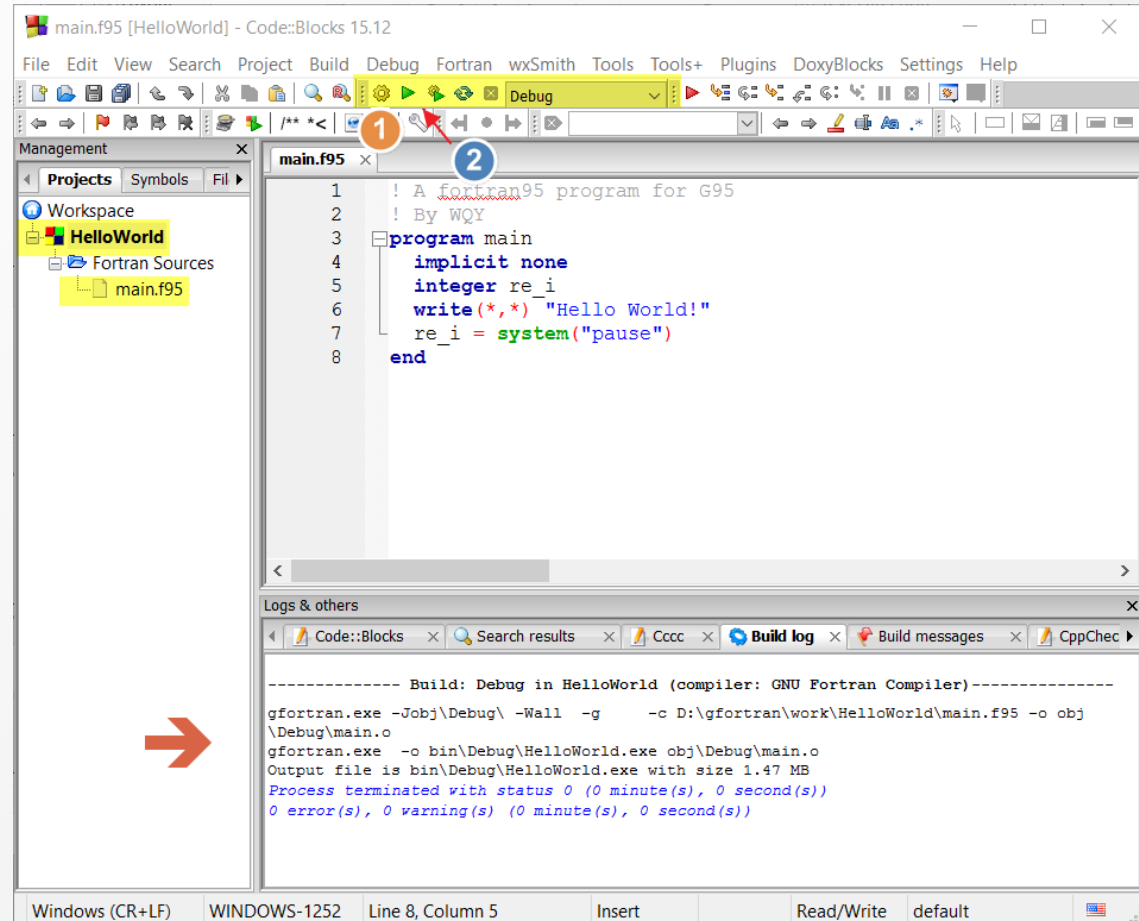
The screenshot displays the Code::Blocks IDE interface. The main window shows the source file 'main.f95' with the following Fortran code:

```
1  ! A fortran95 program for G95
2  ! By WQY
3  program main
4      implicit none
5      integer re_i
6      write(*,*) "Hello World!"
7      re_i = system("pause")
8  end
```

The left sidebar shows the 'Management' pane with 'Projects' and 'Symbols' tabs. Under 'Projects', a 'HelloWorld' project is listed, containing 'Fortran Sources' and the file 'main.f95'. The bottom status bar indicates 'Windows (CR+LF)', 'WINDOWS-1252', 'Line 1, Column 1', 'Insert', 'Read/Write', and 'default'.

# Compile, build and run your first program

- Click on 'build' button to compile and build your program (*step 1*)
- Click on 'Run' button (*step 2*)



The screenshot shows the Code::Blocks IDE interface. The 'main.f95' file is open, displaying the following Fortran code:

```
1 ! A fortran95 program for G95
2 ! By WQY
3 program main
4   implicit none
5   integer re_i
6   write(*,*) "Hello World!"
7   re_i = system("pause")
8 end
```

The 'Build log' window at the bottom shows the output of the compilation and execution:

```
----- Build: Debug in HelloWorld (compiler: GNU Fortran Compiler)-----
gfortran.exe -Jobj\Debug\ -Wall -g -c D:\gfortran\work\HelloWorld\main.f95 -o obj\Debug\main.o
gfortran.exe -o bin\Debug\HelloWorld.exe obj\Debug\main.o
Output file is bin\Debug\HelloWorld.exe with size 1.47 MB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```



# See the result

The screenshot shows the Code::Blocks IDE with a Fortran project named 'HelloWorld'. The main window displays the source code of 'main.f95':

```
1  ! A Fortran95 program for G95
2  ! By WQY
3  program main
4      implicit none
5      integer re_i
6      write(*,*) "Hello World!"
7      re_i = system("pause")
8  end
```

A separate console window titled 'D:\gfortran\work\HelloWorld\bin\Debug\HelloWorld...' shows the output:

```
Hello World!
Press any key to continue . . .
```

The status bar at the bottom of the IDE indicates 'Line 6, Column 1'.



# Some exercises

- Exercises
  - Create a Fortran program with two source files
  - Compile only the source files, one by one using 'compile current file' under 'build' menu item
  - Rebuild your program by using the toolbar icon
  - Try debugging capabilities of CB
    - Add some variables to watch window
    - Step into procedures
  - Try other Fortran projects provided by CB
    - Make a Fortran static library
    - Make a Fortran DLL
  - Try workspace capabilities of CB
    - Create two projects
    - Make one dependent on the other