

CSCU9V4 Practical 1

Getting to Compilation

Introduction

C is more than 40 years old. There are multiple compilers and IDEs. This practical describes but one method to set-up a full working environment in Windows. While the IDE is likely to be the preferred format for most, there are also instructions to configure a text environment.

Note: Compiling at the command line (text environment) allows for use of any text/code editor, at the cost of making some programming tasks less intuitive.

Getting Started

Netbeans (IDE) + MinGW (compiler package) will be the default environments on campus. To avoid confusions in the future, first set C as the default for new projects:

- Launch a new Project
- Select C/C++ in the left pane; now C/C++ Application in the right pane; click Next.
- **Make sure:** To select C from drop-down. This should now be set as the default.
- **Make sure:** To change the *Location* from UNC path, i.e. \\... , to **H:**\. UNC is unsupported in Netbeans.
- Give your project a name, ie. myFirstC, and continue.
- NOTE: Standard practice is to create a new folder for each project, with the same name.

Open `main.c` by double-clicking in the Project pane (left) under 'Source Files'.

Strongly Recommended

Set the `-Wall` compiler option:

- In the Project pane (left), right-click on the project name → Properties.
- Now Build → C Compiler → click 'Additional Options'.
- Type `-Wall` and accept the change.

Getting Familiar

Now take the time to mouse-over and become familiar with the icons in the toolbar. Take special note of all buttons that compile and/or run.

In the Project pane on the left, click the '+' symbol to reveal all of the projects resources (source files, images, etc). There should appear only `main.c` -- Clearly, source files are edited in the large text editing pane in the centre of the screen.

Compile and run your program now. A CLI window should appear, even though it does nothing. The bottom window expands to reveal compiler and linker feedback, as well as errors and warnings if there are any.

Feel free to navigate, investigate, and play around. The worst that can happen is a restart of the IDE!

Do something!

Time to write code! Start with hello world. If choosing to use a text editor + command line instead of an IDE, remember that every C program must at a minimum:

- Include headers; all CLIs will need `stdio.h`
- The `main()` 'driver'. IDEs generate function arguments for program parameters. If none are required, then `void` is sufficient as a parameter.
- Finishing with `return 0` or `equivalent;` is a good idea to let the system know that the program has completed without incident.

C syntax is very similar to Java. Even `printf()` is available in Java, though it is the convention in C. Try to write something – anything! Declarations, basic math, even loops are (almost) identical!