

INF1002 Programming Fundamentals - Labs

Introduction to C programming and development environments

1. OBJECTIVES

To choose, install, and configure a development environment suitable for C programming.

2. CHOOSING A DEVELOPMENT ENVIRONMENT

Unlike Python, C is not packaged with a standard development environment like IDLE. Instead, many different vendors have created C compilers for many different computing platforms, and an even greater number of vendors have created editors for C programs (these editors usually support many other languages as well). You may choose whichever compiler and editor that you like. This section describes a few well-known options that you can access for free.

2.1. WINDOWS

There are a few main free options for C development on Windows, e.g. Microsoft Visual Studio, Code::Blocks, etc. There also numerous programmers' editors with support for C such as Notepad++, UltraEdit, and Atom, which can be used with a command-line compiler.

2.2. MICROSOFT VISUAL STUDIO

Visual Studio is popular amongst professional programmers and has many features designed to make working on larger programs easier but can be resource-intensive and overwhelming at first. You can install the Community Edition for free on your own computer by visiting <https://visualstudio.microsoft.com> and following the instructions.

- (i) The "Getting Started" page at <https://visualstudio.microsoft.com/vs/getting-started/> provides several tutorials intended to introduce you to Visual Studio. Unfortunately, it is not obvious how to create a C program in Visual Studio even though it otherwise has good support for the language: you need to create an empty Visual C++ project (i.e. "Empty Project" under "Visual C++ > General" when creating a new project), then rename all your source files to have a ".c" extension instead of the default ".cpp".
- (ii) (ii) If you want to use Microsoft's compiler, but would rather use another editor, you can also access the command-line compiler ("cl") using the "Developer Command Prompt" available in the Start menu. The Developer Command Prompt is just normal command prompt with its environment set up to use the compiler tools, so that you can compile a C program called *hello.c* with
cl hello.c

This will create an executable file *hello.exe*, which can be run from the command line as usual. More complex programs made up of multiple ".c" files require some more complex operations, but this pattern should be enough to get us through today's lab.

2.3. MAC OS X

Apple provides a development environment for MacOS X called Xcode, which you can

download from <https://developer.apple.com/xcode/> (you'll need to have an Apple ID). Xcode includes the GNU C Compiler, a source code editor, and numerous other features that we do not need for this course.

2.4. LINUX

Most Linux distributions have the GNU C Compiler (*gcc*) already installed. If not, use your package manager to install it. Most distributions also come with a text editor with some support for C programming, such as *gedit* (for GNOME-based desktops) or *Kate* (for KDE-based desktops). Many other editors exist if you want to try them.

To compile a program called *hello.c* from the command line, execute

```
gcc -o hello hello.c
```

This compiles and links *hello.c* to create an executable file called *hello*. You can then execute the program using

```
./hello
```

2.5. CROSS PLATFORM: CODE::BLOCKS

Code::Blocks is a *free cross-platform IDE for C, C++ and Fortran* built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable. You may download it from <http://www.codeblocks.org/> and install it on your computer. You may also learn how to use it at: <http://www.codeblocks.org/user-manual>

2.6. ONLINE

You may also use on-line compilers like

<https://repl.it/>

<https://pythontutor.com/c.html#mode=edit>

to test out your C programs.