SUMMER PARTICLE ASTROPHYSICS WORKSHOP 2023

May 10

Getting Ready to Get Started with C++

Jonathan Hucker

1 Installing a Compiler

So you've decided to embark on the adventurous journey of learning C++. Great! A saying I try to live by, "the more you do the less you wait", embodies the nature of C++ compared to dynamic programming languages such as Python, Perl, and JavaScript. Once compiled, you obtain an executable file that runs extremely fast. There are two costs to achieve this benefit. The first: there is more we as programmers must do to ensure the script compiles. The second: it is a high level programming language (really a mid level language), we must first build/compile the code with a compiler. To ensure you are prepared for C++ seminar, you are responsible for ensuring your machine has a compiler capable of running C++ 11. This document provides resources for obtaining a working compiler for windows, mac, and linux. The steps referenced in the resources have been tested and provide a straightforward method for installing the GNU Compiler Collection (GCC).

1.1 Windows

The following instructions have been tested on a Microsoft Windows 10 OS, version 10.0.19044, and successfully installed Minimalist GNU for Windows (MinGW), which for our purpose will serve well as a c++ compiler. The version of gcc installed by this method is gcc 6.3.0. This installation also gives you the option of installing not only a c++ compiler, but an ADA and Fortran compiler as well (if you so wish).

First, open a terminal window by navigating to the search bar and typing *Command Prompt*. To see if you have already have gcc installed and set up on your system, type *gcc -v*. This command will tell you some various specs including the gcc version you have installed. If you do not have gcc installed (or have the path set incorrectly, more on this later), then the system returns *'gcc' is not recognized as an internal or external command, operable program or batch file.*

To set up gcc, follow the steps provided and continue until you complete step 9. Not that step 6 is a bit strange, but yes, you close the installation manager at this stage to continue with the install.

Install link → Windows ← Install link

One you have completed up to and including step 9, you will have successfully installed gcc on your system. But if you try and compile a C++ file or type *gcc -v*, you will likely receive the *'gcc' is not recognized as an internal or external command, operable program or batch file* message again. To remedy our situation, we need to adjust the system environment.

- Navigate to the search bar or push the windows key, type *environment*, and select *Edit the system environment variables.*

- Select *Environment Variables...*

- Select the *"Path"* option under *"System variables"* and click Edit....

- Type *C:\MinGW \bin* into the field and click ok, then ok, then ok.

If you now head back to your terminal and type *gcc -v*, your system should now return some information about your compiler. To be absolutely sure that your compiler can successfully compile a C++ script, download the YouDidIt.cpp file. In the terminal, navigate to where the file is stored (probably your Downloads folder), recall you can do this using *cd Downloads* if you are in your home folder. Once in the appropriate folder where the YouDidIt.cpp file is stored, type *g++ YouDidIt.cpp* into the terminal. It appears nothing has happened, but that action has successfully compiled the program. You can now run the executable file generated by the program by typing *a.exe* in the terminal. You are now ready for the seminar!

1.2   MacOS

Admittidly, my knowledge of Mac is somewhat limitted. This video MacOS tutorial appears to be quite useful in installing gcc and compiling with g++ with MacOS. These steps have been tested and appear to work okay.

To determine if the compiler works, let's try and run some C++ code. Download the file named YouDidIt.cpp and store it in your Downloads folder. Open a terminal and navigate to your downloads folder using the *cd* command (i.e., *cd Downloads*). In the terminal, type *g++ YouDidIt.cpp*. If your compiler is successful, it will appear as if nothing has happened. In reality, the compiler has succesfully compiled your script and has created the executable file "a.out". To run this file, simply type in the terminal *./a.out*. You are now ready for the seminar!

1.3   Linux

Typically the gcc compiler comes preinstalled with most linux machines, such as Ubuntu 20.04 and beyond. If you are unsure if you have gcc, navigate to your search bar and open a terminal. In the terminal type *gcc -v*. If you have gcc, the system will return with information on your version of gcc. Otherwise, you will need to install the compiler.

Fortunately, this is a simple task. In your terminal type the following commands:

- sudo apt update

- sudo apt install build-essential (it may ask you to type *y* to continue the install

- gcc -v

⁸²   While typing *gcc -v*, your system should now return some information about your compiler. To be
⁸³ absolutely sure that your compiler can successfully compile a C++ script, download the YouDidIt.cpp
⁸⁴ file. In the terminal, navigate to where the file is stored (probably your Downloads folder), recall you
⁸⁵ can do this using *cd Downloads* if you are in your home folder. Once in the appropriate folder where the
⁸⁶ YouDidIt.cpp file is stored, type *g++ YouDidIt.cpp* into the terminal. It appears nothing has happened,
⁸⁷ but that action has successfully compiled the program. You can now run the executable file generated
⁸⁸ by the program by typing *./a.out* in the terminal. You are now ready for the seminar!
⁸⁹

1.4   A Last Resort

⁹¹   If you have been having difficulties installing a C++ compiler on your machine, there is a last resort
⁹² option: in browser compiler: online compiler. This option should really only be used for practicing C++.
⁹³ If you plan on coding in C++, its is best to have your own compiler installed on your system. For our
⁹⁴ purposes however, this option is okay.
⁹⁵