PHYS4070 Quick start guide

This will guide you through getting set up, and compiling your first basic c++ program.

You have two basic options:

1. install a c++ compiler (and any other tools we'll need later on) on your own computer/laptop
2. Use the smp-teaching server

If you use your own PC, things might be a bit quicker/nicer, and you'll be able to work on your assignments without needing an internet connection. The downside is that sometimes getting eveything working, particularly on windows, can be difficult (though we are happy to help you out!) Using the smp-teaching server might be a little slower, but eveything you'll need is already installed there ready-to-go.

We recommend you use Visual Studio Code https://code.visualstudio.com/ to edit your code. It's available on school lab PCs, and is free to download.

---

1. Install c++ compiler on your own computer [optional]

linux

- Install the compiler using terminal:
- e.g., on ubuntu: `sudo apt install g++`
  - this will install the standard GNU c++ compiler
- Install LAPACK library: `sudo apt install liblapack-dev libblas-dev`
  - This installs the basic linear algebra libraries (for solving matrix/eigenvalue problems etc.)
- If you will use GNUPLOT for plotting: `sudo apt install gnuplot`

MacOS

Install homebrew (a linux-like package manager) - install from "https://brew.sh/"

Then, using the terminal

- install g++ using: `brew install gcc`
- install LAPACK: `brew install lapack`
- If you will use GNUPLOT for plotting: `brew install gnuplot`

Windows

- Easiest way is to use WSL (Windows subsystem for linux)
- To get this working:
  - Press Windows key + R then type: **optionalfeatures.exe** then hit Enter (or search for "turn windows features on or off" in the start menu).
  - Scroll down then tick "Windows Subsystem for Linux" then click OK. You'll have to re-start your pc after this
  - You can then install ubuntu through the regular windows app store - then continue as per the linux instructions above
  - Full Instructions + more details here: docs.microsoft.com/en-us/windows/wsl/
- Once WSL is installed, you can access all your windows files via directory: /mnt/c/ (this is the C:\ directory)
  - I recommend creating a folder for phys4070 in your documents, e.g., C:\Users\username\Documents\PHYS4070\
  - Then, from WSL, navigate to it like: `cd /mnt/c/Users/username/Documents/PHYS4070/`
  - You can create a symbolic link to this if you like, to save typing: `ln -s /mnt/c/Users/username/Documents/PHYS4070`
  - Then, you can use windows tools (like visual studio code) to write the code in windows, and use WSL to compile and run it from a terminal
  - VSCode works very well with WSL - there's a 'Remote WSL' plugin you can install. Then, you use VSCode as normal, and let WSL compile/run your code in the background

---

---

2. Use smp-teaching: terminal and/or GUI

School teaching server: *smp-teaching.smp.uq.edu.au*

You should have been given access to the smp-teaching server.
If you don't yet have an account (i.e., you cannot log on), email *help@its.uq.edu.au* to request access, and include 'SMP - smp-teaching server access PHYS4070' in the subject. (Please, ask us first if you have any questions/problems before emailing the busy IT team)

2.1 Remote Deskop GUI Recommended Method

Use a remote desktop service to log on to the smp-teaching server with a GUI (graphical user interface).

The one we recommend is **Visual Studio Code** (note this is different from 'visual studio', becuase naming is hard).

You can also use '*Microsoft Remote Desktop*' program or similar, which is available on both windows and mac if you must.

- Visual Studio Code is already installed on lab PCs. It's free to download: https://code.visualstudio.com/
- Make sure the 'Remote SSH' extension is installed
- In bottom right, click 'open a remote window'. (If you don't see it, use 'search')
- Choose 'Connect to host' and ass a new ssh host: username@smp-teaching.smp.uq.edu.au
- Click 'connect' (bottom right)- and you're ready to go!
- Open a 'Terminal' from inside VSCode and have a look around

2.2 Secure Shell (ssh)

If you don't like fun, you can use the terminal directly. The most basic way is to use ssh (secure shell).

From the command line on linux/mac, type:

- `ssh uqusername@smp-teaching.smp.uq.edu.au`
    - replace 'uqusername' with your username, and enter your regular uq password when prompted
- From windows: you can do the same from a wsl terminal, or install putty (an ssh client for windows)
    - see Windows instructions above for ssh

---

3 Using smp-teaching server important information

The smp-teaching sever runs Cent OS - a free opensource linux distribution. it has all the tools we'll need pre-installed and ready to go.

- The default version of g++ on CentOS 7 is 4.8.5, which quite old, and doesn't support the c++-14 standard. There is a newer version installed, which you need to activate using the command:
  `source scl_source enable devtoolset-8`
  You will want to run this command every time you login to smp-teaching, so you might to add this line to the end of your `.bashrc` file, located at `~/.bashrc`, so that it gets run automatically. (nb: c++-11, which will work by default, may indeed be enough for everything we need)

3.1 Connecting to smp-teaching server from outside UQ

If you are not at UQ and want to access the smp-teaching server, you first need to create a VPN (Virtual Private Network) tunnel to UQ:

- https://www.its.uq.edu.au/services/vpn

and follow the instructions relevant for your operating system. You need to install the VPN software only once but you need to connect each time before you can connect to the CentOS server. ***Don't forget to disconnect the VPN after you are finished since otherwise your whole internet traffic will run through UQ***

3.2 How to transfer files to/from smp-teaching server

1. Use a graphical program, e.g., filezilla [https://filezilla-project.org/]

- Graphical interface lets you transfer files to/from any remote server

2. Use scp (secure copy) from the command-line using linux, mac, or windows (via wsl). Some examples:

- Copy the file "foobar.txt" from a remote host to the local host;
  - `$ scp username@smp-teaching.smp.uq.edu.au:foobar.txt /some/local/directory`
- Copy the file "foobar.txt" from the local host to a remote host
  - `$ scp foobar.txt username@smp-teaching.smp.uq.edu.au:/some/remote/directory`
- Copy the directory "foo" from the local host to a remote host's directory "bar"
  - `$ scp -r foo username@smp-teaching.smp.uq.edu.au:/some/remote/directory/bar`
- More examples: http://www.hypexr.org/linux_scp_help.php

3. Use git, and a git server (github, or the schools git server)

4. Check to see if everything is working

```
#include <iostream>

int main(){
  std::cout << "Hello world!\n";
}
```

- Try to compile the simple 'hello world' example
- Add the example 'hello.cpp' to your working directory, or reproduce the above code yourself
- Compile the code:
    - `$ g++ hello.cpp -o hello`
    - This tells g++ to compile the program in 'hello.cpp'. the '-o hello' tells the compiler to name the output executable 'hello'
- Run the newly compiled code:
    - `$ ./hello`
    - The result should be "Hello world!" printed to the command line
- If this doesn't work, review the above instructions, and if still having problems, contact one of the lecturers or the tutor.

Basic c++tutorials

There are many great websites online for learning c++ (as well as the books recommended on Blackboard). I recommend hackingcpp.com/ - which has a large array of nicely formatted tutorials, examples, and cheat-sheets. Start with

- hackingcpp.com/cpp/beginners_guide.html

For learning C++, we recommend:

- Dmitrović, S. (2020). Modern C++ for Absolute Beginners: A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards (1st ed.).
  The full text is available from the UQ library
  https://search.library.uq.edu.au/permalink/f/tbms52/TN_cdi_askewsholts_vlebooks_9781484260470

There are two main websites for C++ reference material:

- <cppreference.com> - The "standard" most complete/thorough resource, not very beginner friendly
- <cplusplus.com> - Less detailed than the above, but more beginner friendly, and has many nice examples

Compiler Explorer (godbolt.org) is an online compiler that lets you easily test/share snippets of code.

- http://godbolt.org
- Here's an exmaple "in action": https://godbolt.org/z/8Y7Tohj77

---

5. Basics of linux command line

A basic tutorial can be found here:

- https://ubuntu.com/tutorials/command-line-for-beginners
- We'll only really need up to section 5
- This tutorial is based on ubuntu - but everything is the same for any flavour of linux, including the CentOS that runs on smp-teaching.
- Essentially everything is also the same on MacOS

A handy list of common command-line commands:

The manual pages on the computer can be accessed by typing man and the command name, eg. “man pwd”. This tells you what each command does, good to do before you try each of the following
commands:

Syntax	Description
pwd	prints current directoty [Print Working Directory]
ls	lists all files/directories in current directory
mkdir	creates a new directory
cp	copy a file
mv	move (rename) a file
touch	creates a new (blank) file
rm	deletes a file (be **very** careful, there is NO recylce bin)
less	prints the contents of a file to the terminal
cd	change directory

Syntax	Description
.	a '.' is short-hand for current directory
..	short-hand for parent (one-up) directory
~	shortcut for your home directory
g++	runs the g++ compiler
gnuplot	opens plotting program
	from inside gnuplot, try (e.g.) `gnuplot>plot sin(x)`
	type 'quit' or 'exit' to close gnuplot
`tab`	he tab key is very useful - it is autocomplete. It will save you from typing long file names, and will help avoid making typos!

In a Terminal try typing in the following commands in order, and see if you can follow what is hapenning and why. If you don't understand what they are doing, please ask one of us for help:

```
pwd
ls
mkdir temp
ls
cd temp
pwd
ls
touch hello.txt
ls
cp hello.txt hello2.txt
ls
cd ..
pwd
cd ~
pwd
```