

**School of Electrical Engineering and Computing  
University of Newcastle**

**SENG1120/6120 Data Structures – Semester 1, 2021  
Lab 1 (Week 1)**

---

In this lab you will use Cygwin to write, compile and execute your first C++ program.

**1. Cygwin:**

1. Open Cygwin (Start->All Programs->Cygwin)
2. A terminal window should appear. Type `pwd` and press <enter>. This will show you your current 'home' folder.
3. *We recommend using your U: drive however – this way you can save your code on your U: drive, and using the following, you can also direct Cygwin to use this location as your 'home' folder. To do this with your U: drive, simply enter:*

```
cd /cygdrive/u
```

If you are using your own Laptop or PC, you can set this to your own C: drive:

```
cd /cygdrive/c
```

4. Create a subdirectory called **SENG1120** on the terminal window using the command `mkdir SENG1120` - move to that directory using the command `cd SENG1120`.

*In Blackboard, you will find a post in Course materials describing common Unix commands you can use within Cygwin.*

5. Create a subdirectory called **lab1** on the terminal window using the command `mkdir lab1`. Move to that directory using the command `cd lab1`. Go to Blackboard, course materials, computer lab section, and save `test.cpp` onto the folder.

6. Test that compiling the program works using the command

```
g++ -o myCode test.cpp
```

Use `ls` to see that you have created a new file called `myCode.exe`.

7. Run the program using the command `./myCode.exe` – alternatively you can run the same program using just `./myCode` (you don't NEED to specify `.exe`)

***That's it! You have compiled and run your first C++ code in Cygwin.***

## 2. Using Cygwin with the files supplied on Blackboard:

1. Copy the file `add.cpp` for Laboratory 1 from the SENG1120/6120 Blackboard site (in Course Documents ->Laboratories) into the folder for SENG1120 in your Cygwin personal area.

You will note that is you're using your U: drive and completed section 1, you will be able to see and use these folders in Windows Explorer also!

2. You can use the supplied `makefile` to compile other programs. For example, issue the command `make add` and observe that the program defined in the file `add.cpp` is now compiled. Use the command `make clean` to remove any ".o" or "core" files in your directory. In case you were unaware, `core` files are memory dumps produced by some operating systems when a program crashes – something that happens from time to time when you are developing software.

## 3. Your own program:

1. During the lecture there was a example for creating a small temperature conversion program, from Degrees Celsius to Fahrenheit? *Maybe Dan did it, or maybe he did a different example?* But for your own first program, go back and look at the Degrees Celsius to Fahrenheit example - code it, compile it, and run it.

Just to help, to convert between Degrees Celsius and Fahrenheit, use:

$$f = 9/5 * c + 32$$

...where *f* is the temperature in *Fahrenheit* and *c* is in *Celsius*.

You can use the file `add.cpp` as a start point and then add your code to the method `main()`.

2. Does this program work as expected? Why not?

Why does the above formula produce invalid results, but produced correct results when coded as:

$$f = 9.0/5.0 * c + 32.0$$

**Good Luck!**

More info - go to <https://www.youtube.com/watch?v=DtUTe4Xk--8> and watch the video tutorial. It goes through all steps for this lab, except the temperature calculation program; note that step 3, 4, and 5 may vary slightly – **Note: these video's while still highly relevant are to be considered deprecated, due to age.**