Edit/compile/execute

- typical development sequence:
- use editor to write code in a high level language, store in a file
- use compiler to attempt to translate file contents to machine code, store results in another file
- edit HLL code to fix errors the compiler finds, then compile again
- once it compiles cleanly we can try running the executable file
- identify any behavioural errors, edit the HLL code to fix them, then compile again, test again

Compiling C++ on our system

- using any editor we like, create or modify text files containing C++ code ... file extension like .cpp used to indicate it's a C++ file, e.g. myprogram.cpp
- be sure to save our changes, then use the g++ compiler to read the high level code and store a translation in another file, the executable
- g++ myprogram.cpp -o myprogramx
- (note linux doesn't require extensions, I'll often put an x at the end of an executable name to indicate it's executable)

Compilation continued

- If the compiler finds syntax (i.e. grammatical) errors then it will spit out a (sometimes cryptic) error message
- we edit the .cpp file again, save, and recompile
- repeat this until it compiles cleanly
- now we can try running the executable, e.g.

```
./myprogramx
```

- if it doesn't behave as desired, then repeat the edit, save, compile, test process until it *does* behave as desired