Getting started with C

- The book lists the **C C**ompiler as **`cc`**, which is historically accurate
  - If you run the program `cc` in the COMP211 container, it is an alias of `gcc`
  - The g stands for GNU which is an acronym for GNU's Not Unix...
  - GNU is free, open-source software developed as a substitute for proprietary Unix

- For now, to compile examples from the book:

```
learncli$ gcc -g -Wall -std=c11 <filename>.c
```

  What *are* those argument flags?

  - -g - enables debugging symbols
  - -Wall - enables all compiler warnings
  - -std=c11 - compile with the latest C standard expectations

- By default, and historically, this produces an executable binary named a.out
  - To *run* a.out, you have to prefix it with a **./**:

```
learncli$ ./a.out
```

- The *bash* shell's grammar allows you to conditionally run subsequent commands if the prior command exits successfully:

```
learncli$ <command> && <command>
```

  To compile and then run:

```
learncli$ gcc -g -Wall -std=c11 <filename>.c && ./a.out
```

- Infinite loop? Program not responding? Interrupt execution with **`Ctrl+C`**