Programming style

Victor Eijkhout, Harika Gurram,
Je'aime Powell, Charley Dey

Fall 2018

Programming: you, your program, your grade

- It's a necessary condition for your program to compute the right thing.

- But programming style is also important: we may subtract points for an 'ugly' program.

- We will read your source. Make sure it is written in such a way that we can understand it.

- It's all for your own good: badly written code is hard to debug, hard to understand for your colleagues or even yourself half a year from now.

As Donald "I am not worthy" Knuth puts it

> *The best programs are written so that computing machines can perform them quickly and so that human beings can understand them clearly. A programmer is ideally an essayist who works with traditional aesthetic and literary forms as well as mathematical concepts, to communicate the way that an algorithm works and to convince a reader that the results will be correct.*

Abstraction is the name of the game

A good program has a layered design:

- You use the language features to make a set of tools (sometimes called a 'Domain-Specific Language') that talks about your application.
- You write your model or solve your problem using those tools.
- ⇒ You program should read like a story about your problem.

Clean coding principles

- Use meaningful names. (Try a wordcloud on you program.)
- Code should be clear:
  Keep It Simple, Student!
- Program units should have one clear function:
  Single Responsibility principle.
- Any functionality should be implemented only once:
  Don't Repeat Yourself

Language features

Just because a language has a certain feature, does not mean you need to use it.

- Being too clever may give hard-to-read code. May even make your code slower.
- Some C++ features are really from C: no longer needed.
- The four-letter word that starts with g should never be used.

About using the internet

- Yes, you can find solutions on the internet.
- No, that will not make you understand what you're doing.
- We are giving you a sequence of building blocks: try not to use things that you haven't been taught.

TACC