

# Programming style

Charlie Dey, Virginia Trueheart

Spring 2020

# Programming: you, your program

- 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.
- Your source will be graded. 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 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.*

# Programming languages are about ideas

*A powerful programming language serves as a framework within which we organize our ideas. Every programming language has three mechanisms for accomplishing this:*

- *primitive expressions*
- *means of combination*
- *means of abstraction*

*Abelson and Sussman, The Structure and Interpretation of Computer Programs*

# Abstraction is good

Abstraction means your program talks about your application concepts, rather than about numbers and characters and such.

Your program should read like a story about your application; not about bits and bytes.

Good programming style makes code intelligible and maintainable.

(Bad programming style may lead to lower grade.)

# Trust him, he's Dutch!

*The competent programmer is fully aware of the strictly limited size of his own skull; therefore he approaches the programming task in full humility, and among other things he avoids clever tricks like the plague*

— Edsger Dijkstra

# Language features

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

- Being too clever may result in 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.
- Also: we teach you 'modern' C++ and Fortran. Solutions on the internet may use outdated mechanisms.



# Exercise 1

Do an online search into the history of computer programming. Write a page, if possible with illustration, and turn this into a pdf file. Submit this to your teacher.

# Turn it in!

- First go to `portal.tacc.utexas.edu` and create an account.
- Send a piazza personal message to the instructors with:  
your Name, EID, TACC account.
- Wait a little while...
- Use `scp` (or an equivalent) to copy your writeup to your account at `isp.tacc.utexas.edu`:  
`scp yourstory.pdf yourname@isp.tacc.utexas.edu:`
- Ssh to `isp`:  
`ssh yourname@isp.tacc.utexas.edu`  
and use a script to submit it:  
`sdshistory -s yourstory.pdf`  
where the `-s` flag stands for 'submit'.