

---

# CSE 374

# Programming Concepts & Tools

Hal Perkins

Spring 2022

Lecture *finis* – Course Wrapup

# Reminders

---

- Please fill out course evaluations before they close this weekend
- Please nominate great TAs for the Bander award. Thanks.
  - Both for CSE 374 and for other CSE courses
- Final exam: Tue. June 7, 2:30-4:20, regular room (CSE2 G01)
  - Topics: mostly C and tools (make, git, etc.), homework, some C++, but maybe also some shell scripting, etc.
    - No heavy-duty regexp/sed, but you should still know how to use Linux commands and utilities. 😊
  - Closed book but ok to bring two 5x8 hand-written notecards or equivalent
  - Review session Mon. June 6, 4:30-~5:30, also CSE2 G01
    - Bring questions!
- Ed postings: please use descriptive topics! (not just “15su #7”)

# Studying for the exam...

---

- Review *first*, make notes
  - Review lecture slides, sample code, demos, assignments, other resources, etc.
  - Look at topics on the calendar to check your coverage and help organize
  - Brainstorm and trade ideas with colleagues
- “Simulate” an old exam
  - Do it in one timed sitting
  - Working problems is far more important than reading old answers!
- “Grade” yourself, then go back and review problems
  - If still unsure why, ask the staff or your fellow students
  - Rinse and repeat!
- Now, let’s take a look back at the quarter...

# But first...

---

A huge thanks to the folks who made the course work:

Xinyue Chen  
Maxim Klyuchko  
Ben Soesanto  
Dixon Tirtayadi  
Qingyuan Dong

There's a lot of complexity behind the scenes and without a great staff it can't happen.

# And thanks to

---

## You

It has been a tough year with the continuing pandemic. We've been in it for the long haul for a while now. You should be proud of your resilience and what you've done. Please take care of yourself, watch your health, stay active, help yourself, your friends, your community.

# 10 weeks ago....

---

- We have 10 weeks to move to a level well above novice programmer:
  - Command-line tools/scripts to automate tasks
  - C programming (lower level than Java; higher than assembly)
  - Tools for programming
- That's a lot!
- Get used to exposure, not exhaustive investigation
  - This is not intro programming anymore

# What is CSE 374?

---

- Something of a “laundry list of everything else”, but...  
*There is an amorphous set of things computer scientists know about and novice programmers don't. Knowing them empowers you in computing, lessens the “friction” of learning in other classes, and makes you a mature programmer.*
- The goal is to give you a sense of what's out there and what you can expect – and how you can learn more later when you need to

# A few General Areas

---

1. The command line
  - Text-based manipulation of the computing environment
  - Automating (scripting) this manipulation
  - Using powerful *utility* programs
- Let the computer do what it's good at so you don't have to!
- We'll use Linux (an operating system) and bash (a *shell*)
  - but the concepts are not tied to these
- Idea: Knowing the name of what “ought to exist”
- Idea: Programming in a language designed for interaction



# A few General Areas

---

## 2. C (and a little C++)

- “The” programming language for operating systems, networking, embedded devices, ...
- Manual resource management
- Trust the programmer: a “correct” C implementation will run a program with an array-bounds error and whatever happens, happens
- A “lower level” view of programming: all code and data sits together in a “big array of bits” (er, bytes)
- Idea: Parts look like Java – don’t let that deceive you!
- Idea: Learn to think before you write, and test often

# A few General Areas

---

3. Programming tools – so far you have written programs and run them. There are programs for programming you should know about:
  - Compilers (vs interpreters)
  - Debuggers
  - Linkers
  - Recompilation managers
  - Version-control systems
  - Profilers
  - ...

# A few General Areas

---

4. Software development concepts – what do you need to know to write a million lines of code?
  - Testing strategies
  - Team-programming concepts
  - Software specifications and their limits
  - ...

# Perspective

---

“There is more to programming than Java methods”

“There is more to software development than programming”

“There is more to computer science than software development”

You’ve got a lot more in your toolkit than you had 10 weeks ago

Have fun building great things!

And stay in touch! I’d love to learn what you’ve been up to after  
CSE 374!



*That's all Folks!*