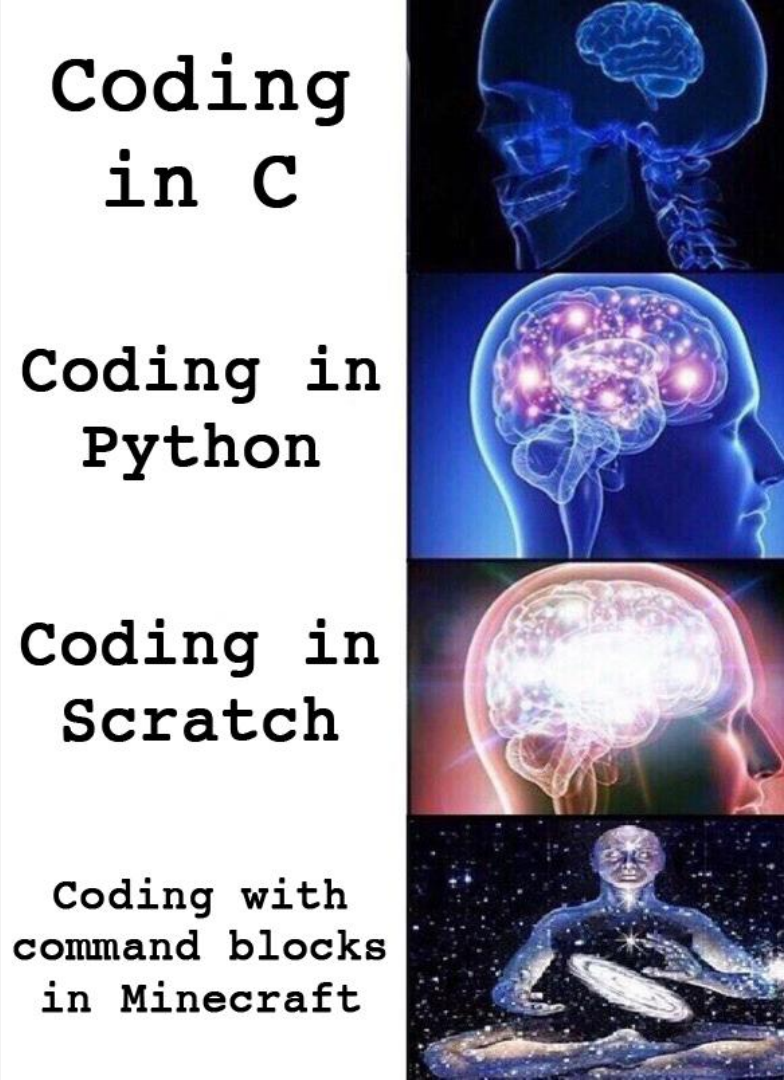
CSc 352 Intro

Benjamin Dicken



Welcome to CSc 352!

Systems Programming and UNIX

C Topics

- General C syntax / language (types, loops, ifs, functions, etc)
- The stack and the heap
- Memory management, malloc, free
- File I/O
- Implementing data structures
- How to debug programs with **GDB**
- How to check for memory leaks with **Valgrind**
- Building with **Make**

UNIX / bash topics

- Files and the file system
- Processes
- General BASH usage (BASH = “Bourne Again SHell”)
- Text processing, regex
- BASH Scripting

Be Prepared

- This is not an easy class
- Be prepared to spend a lot of time, especially on the PAs
- The C stuff will be important to learn for the systems 400-level upper divisions
  - 422, 452, 453, etc
- The bash / unix stuff should be generally valuable for your school *and* professional career

The Instructor

- Benjamin Dicken (Instructor of record)
  - Office: Gould-Simpson 850
  - Email: bddicken@arizona.edu
  - Office Hours
    - See the class website
    - Or by appointment

Teaching Assistants

- **Mahdi Rahimi**, **Kartikey Shukla, Michael Arroyo, Amir Esmaieeli**
- Office hours, grade assignments, etc.

What does it do?

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
  int x1 = rand() + 100;
  int x2 = 50;
  if (x1 > x2) {
    printf("Greater!\n");
  } else {
    printf("Less or equal!\n");
  }
  return 0;
}
```

General Info

- Prerequisites: CS 210 and 252

Class Website

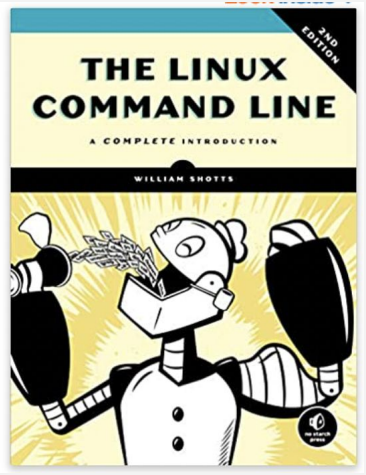
http://benjdd.com/courses/cs352/spring-2023/

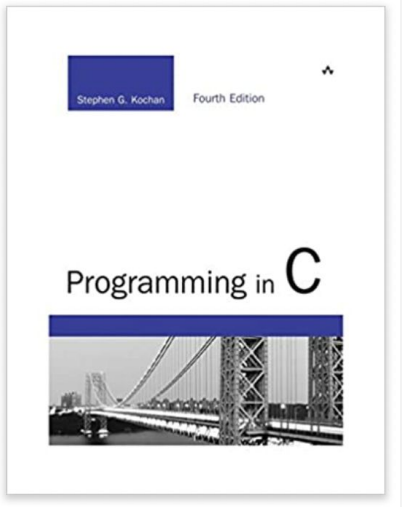
Textbooks

- The Linux Command Line, Shotts (2th)
  - https://www.amazon.com/dp/1593279523/

- Programming in C, Kochan (4th)
  - https://www.amazon.com/dp/0321776410/

- Required readings
- Exams will be open-book for these two books





What contributes to your grade?

- Exams
- Programming Assignments (PAs)
- Quizzes

Activity

How much is each component worth?

Look it up in the syllabus

- Exams
- Programming Assignments (PAs)
- Pop Quizzes

How much is each component worth?

Look it up in the syllabus

- Exams 50%
- Programming Assignments (PAs) 40%
- Pop Quizzes 10%

Exams

- **3 Total**
- First two worth 15% each
  - Open book with the two textbooks
- Final exam worth 20%
- See course schedule for days

Programming Assignments (PAs)

- There will be approximately 10-12 PAs
- Turn in via gradescope
- Compile and run on Lectura
- More about assignment compiling / testing later

Pop Quizzes

- 10 Percent of grade
- 8-15 throughout the course
- Lowest 3 dropped

Grading Policy

- ***Our goal***
  - We will do our best to return grades to you within week of the LATE deadline
- ***If you don't like your grade***
  - You have 5 days from the time your grade is returned to you on Gradescope/D2L/etc to request a regrade. After that, your grade is ***final***

How to get help?

- ***Ask on Discord***
  - Can post to the group chat channels for general questions and guidance.
- ***Direct Email***
  - You’re welcome to email Ben, or one of the TAs
- **Office hours**
  - See class website

Academic Integrity

- When you are working on a PA, you ***can . . .***
  - Talk about ideas and techniques for solving the problem
  - Discuss the spec
  - Talk about the programming at a high-level
- But you may ***not . . .***
  - Share code with each-other
  - Look at each-others code
  - Work on the project together, submit same code
- Exams must be your own original work, no cheating (duh)
- See syllabus, and this

Schedule and Readings

- There will be prep work to go along with each day of the class
- See course schedule

Reading

Go to the class website, and figure out what readings are due for the first week of the course

Sites and Tools

- Sites:
  - Course website - Schedule, Syllabus, Office hour info, PAs
  - Gradescope - PA and Exam grading
  - D2L - Gradebook
  - Discord - Online help and questions
- Tools/software/hardware:
  - Access to Lectura
  - Also, ideally, access to a UNIX computer

The first PA!

- Let’s go to class website

Lectura

- A server provided by the department
- Connect and compile / run your programs from there
- Connect over the internet:
  - Mac / Linux: via **ssh**
  - Windows: via **putty** or **ssh with Windows Subsystem for Linux**

Why lectura? C is not as universally compatible as languages such as Python and Java. A C program that compiles and runs fine on your computer may not on another.

Local and Lectura

- Can install gcc and run C programs locally
- Will need way to transfer to Lectura
  - Mac / Linux: scp
  - Windows: filezilla (or scp / WSL)

Tasks to Complete ASAP

Get this done before next class!!!

Ensure your CS account is set up

If you've not done this before or have forgotten, go to:

https://helpdesk.cs.arizona.edu/

Ensure you can connect to lectura (ssh or Putty)

Do the readings from the TLCL, try out bash on lectura

OPTIONAL: download Windows Subsystem for Linux and try out bash on your own computer

(Windows: linux subsystem)

Lectura Connection Demo

Using Bash and SSH on a Mac

Using Shell / SSH / Putty on Windows

Running a few commands