

CSc 352

Intro

Benjamin Dicken

**Coding
in C**



**Coding in
Python**



**Coding in
Scratch**



**Coding with
command blocks
in Minecraft**



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**
- Build management 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@email.arizona.edu
 - Office Hours
 - See the class website
 - Or by appointment

Teaching Assistant

- Michael Stark
- Will have office hours, grade assignment, etc.

What does it do?

```
#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

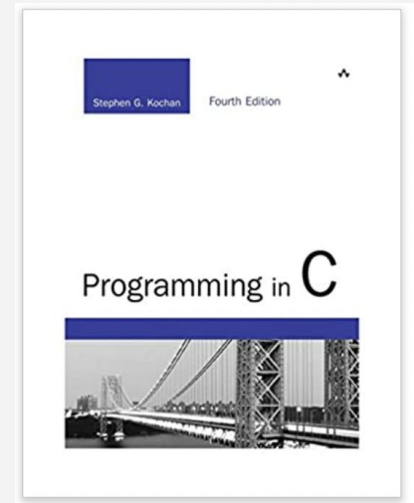
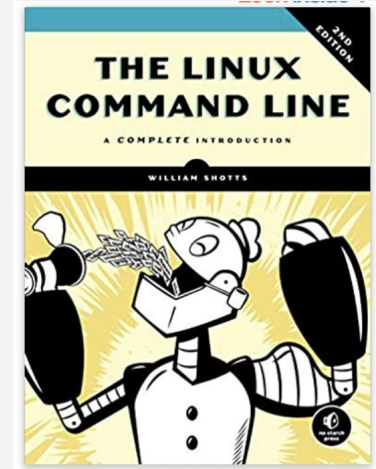
- Prerequisites: CS 210 and 252

Class Website

<http://benjdd.com/courses/cs352/summer-2022/>

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)
- Meetings

How much is each component worth?

Look it up in the syllabus

- Exams
- Programming Assignments (PAs)
- Meetings

How much is each component worth?

Look it up in the syllabus

- Exams 50%
- Programming Assignments (PAs) 40%
- Meetings 10%

Exams

- **3 Total**
- First three worth 15% each (both parts combined)
 - Open book with the two textbooks
- Final exam worth 20%
- See course schedule for days

Programming Assignments (PAs)

- There will be approximately 9 PAs
- Turn in via gradescope
- Compile and run on Lectura
- Will talk more about assignment compiling / testing later



Required Meetings

- Each student required to meet with me / Michael 3 times
 - Beginning → Exam 1
 - Exam 1 → Exam 2
 - Exam 2 → Final Exam
- The first must be with me
- Worth (10/3) % of your grade each

Grading Policy

- ***Our goal***
 - We will do our best to return grades to you within week of the LATE deadline (so long as you turn it in on time)
- ***If you don't like your grade***
 - You have 7 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 and Michael directly
- **Office hours (via Discord)**
 - Office hours will be via discord too
 - Visit these for more specific / direct help with homework

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
- See syllabus, and [this](#)

Exams

- Everything on an exam must be your own, original work
- No copy/pasting, no googling, etc, etc

Schedule and Readings

- There will be prep work (typically reading from the textbook) that go along with each day of the class
- Technically, you *can* get away without doing the readings, but would not recommend
- 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 Office hours
- Tools/software/hardware:
 - Access to Lectora
 - Also, ideally, access to a UNIX computer

The first PA!

- Let's go to class website

Lectura

- A server provided by the department
- Must 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