

```

$ cat welcome.c
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("Welcome to COS 217\n");
    printf("Introduction to Programming Systems\n\n");
    printf("%s %d\n", "Fall", 2020);
    return 0;
}

$ cat Makefile
CC=gcc217
welcome: welcome.o

$ make
gcc217 -c -o welcome.o welcome.c
gcc217 welcome.o -o welcome

$ ./welcome
Welcome to COS 217
Introduction to Programming Systems
Fall 2020

```

1

## Agenda

**Course overview**

- **Introductions**
- Course goals
- Resources
- Grading
- Policies

**Getting started with armlab**

- Brief overview of Linux and bash
- bash walkthrough (separate video)

2

## Introductions

**Lead Instructor**

- Christopher Moretti [cmoret@cs.princeton.edu](mailto:cmoret@cs.princeton.edu)

**Lead Preceptor**

- Xiaoyan Li [xiaoyan@cs.princeton.edu](mailto:xiaoyan@cs.princeton.edu)

**Preceptors**

- Donna Gabai [dgabai@cs.princeton.edu](mailto:dgabai@cs.princeton.edu)
- Scott Karlin [scott@cs.princeton.edu](mailto:scott@cs.princeton.edu)
- Weicong Dong [weicongd@princeton.edu](mailto:weicongd@princeton.edu)
- Juan Duque [duque@princeton.edu](mailto:duque@princeton.edu)
- Ben Kaiser [bkaiser@cs.princeton.edu](mailto:bkaiser@cs.princeton.edu)
- Anne Kohlbrenner [akohlbrenner@princeton.edu](mailto:akohlbrenner@princeton.edu)
- Dale Lee [dalelee@cs.princeton.edu](mailto:dalelee@cs.princeton.edu)
- Huihan Li [huihanl@princeton.edu](mailto:huihanl@princeton.edu)
- Pi Songkuntham [pisong@princeton.edu](mailto:pisong@princeton.edu)

3

## Agenda

**Course overview**

- Introductions
- **Course goals**
- Resources
- Grading
- Policies

**Getting started with armlab**

- Brief overview of Linux and bash
- bash walkthrough (separate video)

4

## Goal 1: Programming in the Large

Learn how to compose large(r) computer programs

**Topics**

- Modularity/abstraction, information hiding, resource management, error handling, testing, debugging, performance improvement, tool support

5

## Goal 2: Lower-level Languages

```

int main(void) {
    while ((iChar = getchar()) != EOF) {
        lCharCount++;
        if (isspace(iChar)) {
            if (iInWord) {
                lWordCount++;
                iInWord = FALSE;
            }
        }
    }
}

```

**THE C PROGRAMMING LANGUAGE**

```

main:
.LFB0:
.cfi_startproc
stp x29, x30, [sp, -16]!
.cfi_def_cfa_offset 16
.cfi_offset 29, -16
.cfi_offset 30, -8
add x29, sp, 0
.cfi_def_cfa_register 29
b .L2

```

RELOCATION RECORDS FOR [.eh\_frame]:

OFFSET	TYPE	VALUE
000000000000001c	R_AARCH64_PREL32	.text

Contents of section .text:

```

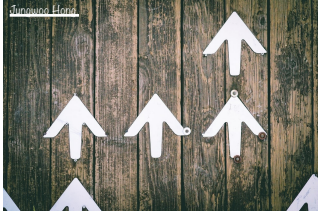
0000 fd7bbfa9 fd030091 39000014
00000090 .{.....9.....

```

6

## Goals: Summary

Help you to gain ...



**Programming Maturity**

7

## Specific Goal: Learn C

**Question:** Why C instead of Java?

**Answer 1:** A primary language for “under the hood” programming in real code bases.

**Answer 2:** A variety of experience helps you “program in the large”



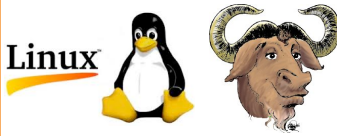
8

## Specific Goal: Learn Linux

**Question:** Why use the Linux operating system?

**Answer 1:** Linux is the industry standard for servers, embedded devices, education, and research

**Answer 2:** Linux (with GNU tools) is good for programming (which helps explain answer 1)



9

## Agenda

### Course overview

- Introductions
- Course goals
- **Resources**
- Grading
- Policies

### Getting started with armlab

- Brief overview of Linux and bash
- bash Walkthrough (separate video)

10

## Lectures

### Lectures

- Describe material at a mix of levels
  - Some conceptual (high) overview
  - Some digging into details
- Slides available via course website
- Videos released TTh on Canvas
- “Watch Party” on Zoom TTh 10am-11am.



### Etiquette

- Watch the lecture before going to precept, otherwise you may end up lost in precept and slow down the rest of the class
- “Watch Party” office hours are for questions about lecture content topics only, not about help with assignments.

11

## Precepts

### Precepts

- Describe material at the “practical” (low) level
- Support your work on assignments
- Hard copy handouts distributed during precepts
- Handouts available via course website

### Etiquette

- Attend your precept: attendance will be taken
  - Must miss your precept? ⇒ inform preceptors & attend another
- Use TigerHub to move to another precept
  - Best for this to happen organically (more than 25% move ≥ 1x)
  - Issues ⇒ See Colleen Kenny (info on website)

**Precepts begin Wednesday/Thursday!**

13

## Websites

<https://www.cs.princeton.edu/~cos217> (Course website)

- Home page, schedule page, assignment page, policies page

<https://princeton.instructure.com/courses/561> (Canvas)

- Links to Zoom precepts, Ed, recorded lectures and precepts, Library reserves and other readings, NameCoach



14


## Ed

**Ed**

- <https://us.edstem.org/courses/2185/discussion/>
- Also available as a Canvas link
- Instructions provided in first precept

**Etiquette**

- Study provided material before posting question
  - Lecture slides, precept handouts, required readings
- Read / search all (recent) Ed threads before posting question
- Don't reveal your code!
  - See course policies



15

## Books

**C Programming: A Modern Approach (Second Edition) (required)**




- King
- C programming language and standard libraries

**ARM 64-bit Assembly Language (required)**

- Pyeatt with Ughetta

**The Practice of Programming (recommended)**

- Kernighan & Pike
- "Programming in the large"

16


## Manuals

**Manuals (for reference only, available online)**

- *ARMv8 Instruction Set Overview*
- *ARM Architecture Reference Manual*
- *Using as, the GNU Assembler*

**See also**

- Linux *man* command'



17

## Help sessions

**Office Hours (starting Wednesday 9/2)**

- 4+ hours every weekday + 2 hours Sunday
- Schedule is on the course website
- Links are on Ed

**LabTAs**

- Your peers are available 4-6 hours per day, every single day
- These are specific to debugging your assignments, for conceptual help with course materials, go to office hours
- <https://labta.cs.princeton.edu/>

18

## Agenda

**Course overview**

- Introductions
- Course goals
- Resources
- **Grading**
- Policies

**Getting started with armlab**

- Brief overview of Linux and bash
- bash Walkthrough (separate video)

19

## Grading

Course Component	Percentage of Grade
Assignments *	66
Midterm Exam **	10
Final Exam **	20
Participation ***	4

\* 6 assignments \* 11% each; penalties for lateness

\*\* During midterms week and final exam period, respectively

\*\*\* Did your involvement benefit the course?


- As measured through precept attendance, precept participation, and Ed participation
- Scaled down from prior terms due to being online

21

## Programming Assignments

Regular (every 1.5-2.5 weeks) assignments

- Introductory survey
- "De-comment" program
- String module
- Symbol table module
- Directory and file trees \*
- Assembly language programs \*
- Buffer overrun attack \*



\*(partnered assignment)

**Assignments 0 and 1 are available now**

**Start early!!**

22

## Agenda

**Course overview**

- Introductions
- Course goals
- Resources
- Grading
- **Policies**

**Getting started with armlab**

- Brief overview of Linux and bash
- bash Walkthrough (separate video)

23


## Policies

Learning is a collaborative activity!

- Discussions with others that help you understand concepts from class are encouraged

But programming assignments are graded!

- Everything that gets submitted for a grade must be exclusively your own work
- Don't look at code from someone else, the web, Github, etc. – **see the course "Policies" web page**
- Don't reveal your code or design decisions to anyone except course staff – **see the course "Policies" web page**



**Violations of course policies**

- Typical course-level penalty is **0**
- Typical University-level penalty is **suspension**

24

## Questions?

26

## Agenda

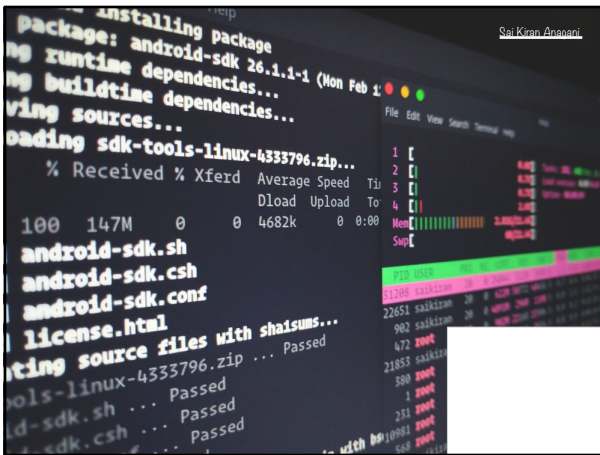
**Course overview**

- Introductions
- Course goals
- Resources
- Grading
- Policies

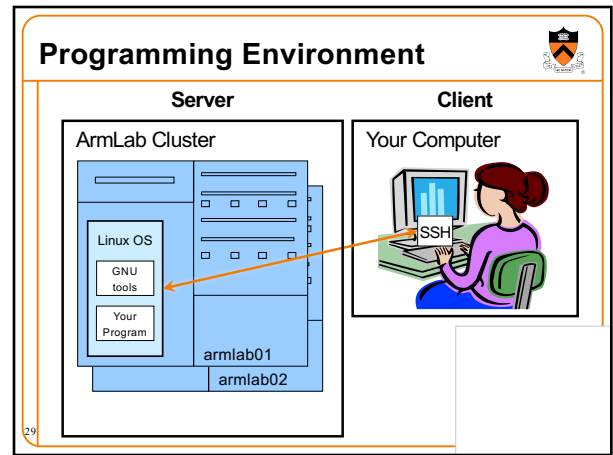
**Getting started with armlab**

- **Brief overview of Linux and bash**
- bash Walkthrough (separate video)

27



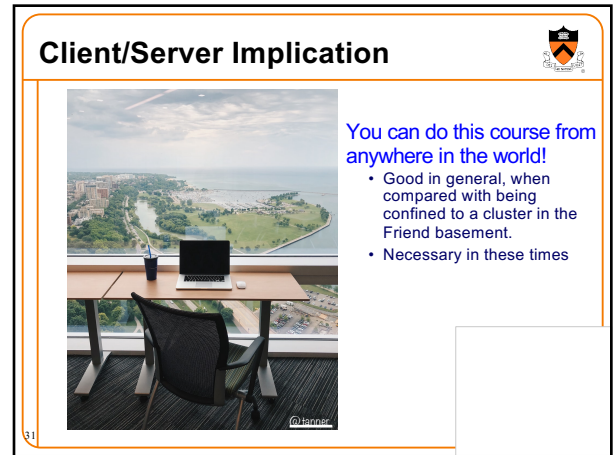
28



29



30



31



32