CS110 Lecture 1: Introduction

CS110: Principles of Computer Systems

Winter 2021-2022

Stanford University

Instructors: Nick Troccoli and Jerry Cain

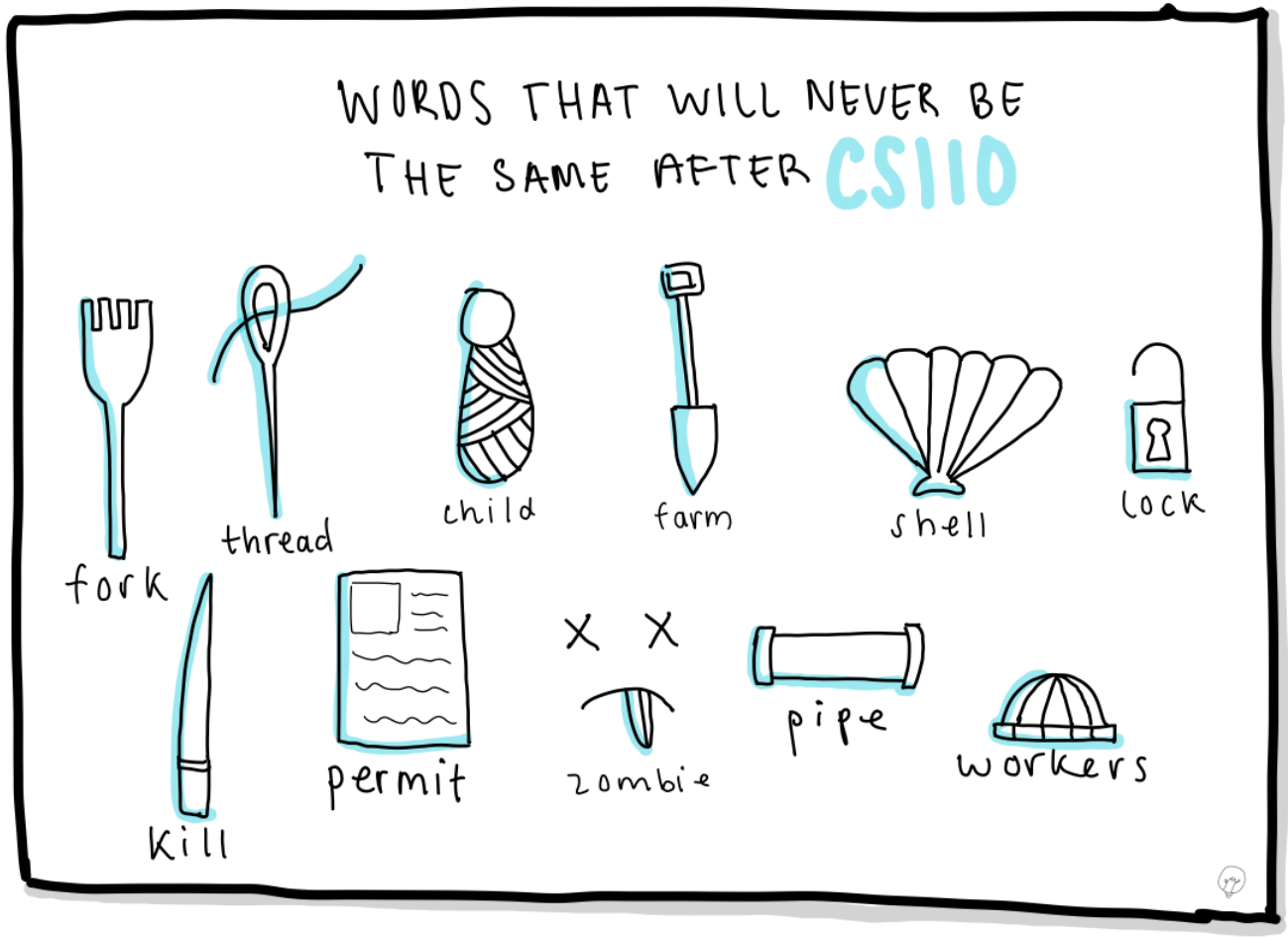

Illustration courtesy of Ecy King, CS110 Champion, Spring 2021

PDF of this presentation

Asking Questions

- Feel free to raise your hand at any time with a question
- If you are more comfortable, you can post a question in the Ed forum thread for each day's lecture (optionally anonymously)
- We will monitor the thread throughout the lecture for questions

Visit Ed (or access via Canvas):

https://edstem.org/us/courses/16701/discussion/

Today's thread:

https://edstem.org/us/courses/16701/discussion/979087

Lecture Plan

- Introduction
- Course Topics Overview
- Course Policies

Lecture Plan

- **Introduction**
- Course Topics Overview
- Course Policies

Guiding Principles For This Quarter

- We are each starting the new year in unique circumstances.
- We are likely not fully recovered or restored from the stresses of the past 2 years and now facing new uncertainties, responsibilities, and emotions.
- We will do everything we can to support you. We have designed the course to the best of our ability to provide flexibility.
- We will constantly evaluate and listen to ensure the class is going as smoothly as possible for everyone.
- Please communicate with us if any personal circumstances or issues arise! We are here to support you.

What is CS110?

CS107 took you behind the scenes:

- how things work inside C++/Python/Java, and how your programs map onto the components of computer systems
- understanding of program behavior and execution

What is CS110?

CS107 took you behind the scenes:

- how things work inside C++/Python/Java, and how your programs map onto the components of computer systems
- understanding of program behavior and execution

CS110 uses this as a foundation to build complex programs that maximally take advantage of the hardware and operating system software available to us:

- How can we understand the designs and tradeoffs of large systems?
- How can we write software that spans multiple machines?
- How can we write software that runs tasks in parallel on a single machine?

Prerequisites

CS107 or equivalent -

- Each of you should know C and C++ reasonably well so that you can...
  - write moderately complex programs (e.g. pointers, **malloc/realloc/free**, C strings, C++ classes, methods, references, templates, **new/delete**)
  - read and understand portions of large code bases
  - trace memory diagrams and always win!
- Each of you should be fluent with **Unix**, **GDB**, **Valgrind**, and **Make** to the extent they're covered in CS107 or its equivalent.

Prerequisites

CS107 or equivalent -

- Each of you should know C and C++ reasonably well so that you can...
  - write moderately complex programs (e.g. pointers, **malloc/realloc/free**, C strings, C++ classes, methods, references, templates, **new/delete**)
  - read and understand portions of large code bases
  - trace memory diagrams and always win!
- Each of you should be fluent with **Unix**, **GDB**, **Valgrind**, and **Make** to the extent they're covered in CS107 or its equivalent.

The first assignment is meant to give you a sense of the scope of CS110 programs and refresh your memory on relevant prerequisites. If you feel ok about it, you're all set!