

Lecture 1a: Welcome!

CS 70: Data Structures and Program Development
January 21, 2020

1

Course Goals

- Data structures
- Practical programming skills
- Deeper understanding of computer memory

More on course wiki... 2

Learning Goals

1. I can identify good and bad pair-programming practices.
2. I can model basic functions in the CS 70 memory model.
3. I can explain the difference between machine code, assembly code, and source code.
4. I can explain the difference between a compiler and an interpreter.

3

Introducing Your CS 70 Professors

Prof. Erin Talvitie

- Olin 1279
- CS Interests: AI, Reinforcement Learning

Prof. Beth Trushkowsky

- Olin 1267
- CS Interests: Databases, Crowd Querying

Prof. Lucas Bang

- Olin 1271
- CS Interests: Quantitative Program Analysis

4

Question: Pair Programming

What are the advantages of pair programming (in CS 70, or generally)?

5

Question: Pair Programming

How can “Pair Programming” skills improve with practice?

6

The Rock Lab (BK B111)



7

CS70 Partners

- 3 different partners over the course of 10 Homeworks.
- Can partner with anybody in any section of CS70.
- **MUST** be able to attend the same Friday lab section as your partner.

8

How does a program run?

In a groups of 4, what do you remember about HMMM
(from CS 5 or CS 42)?

9

CS 70 Memory Model

11

Example: Simple C++ Function

```
int main() {  
    int x = 30;  
    int y = 12;  
    int z = x + y;  
}
```

12

Compiling

- C++ program **compiled** into an equivalent representation for a target computer

13

Bits encode everything

- Encode integers, floating point, strings, instructions...

14

Example: Source Code

fourtwo.cpp:

```
int main() {  
    int x = 30;  
    int y = 12;  
    int z = x + y;  
}
```

15

Example: Assembly Code

fourtwo.s:

(...14 Lines Omitted for Space...)

```
movl    $30, -4(%rbp)  
movl    $12, -8(%rbp)  
movl    -4(%rbp), %ecx  
addl    -8(%rbp), %ecx  
movl    %ecx, -12(%rbp)  
popq    %rbp  
retq
```

(...8 Lines Omitted for Space)

1100 0111 0100 0101
1111 1100 0001 1110

16

As viewed by a text editor!

[illegible]

17

As viewed by a text editor!

[illegible]

18

19

20

Class Exercise: Scenarios

1. Form a group of 4 people
2. Take Scenario handouts.
3. Decide how each of your scenarios is best described:
 - Encouraged
 - Acceptable
 - Discouraged
 - Forbidden

22

1

Students A and B are paired. They try to compile their code, and get the error

```
corroborate.cpp:213:1: error: C++ requires  
a type specifier for all declarations
```

Seeing no obvious problems at line 213, column 1 of `corroborate.cpp`, the students enter

```
"C++ requires a type specifier for all  
declarations"
```

(the generic part of the error message) into Google. The first hit leads them to a Stack Overflow post explaining how someone else encountered and fixed that error. A and B realize their code has the same problem; they fix it, and the error disappears.

23

2

Students A and B are paired. They get out two laptops, sit next to each other, and double their coding speed by editing two different files at the same time.

24

3

A CS 70 homework assignment asks for an implementation of Red-Black Trees. This data structure seemed to make sense in class, but afterwards Student A realizes that some parts still aren't clear. Before starting the homework, A browses the web and reads some other high-level explanations of Red-Black Trees, being careful not to look at detailed implementation discussions or source code.

25

4

Students A and B are paired, and want to add a new character to the end of a string. They realize they need to learn more about the `string` class in C++, so they try the following web search:

`string class C++ standard library`

This leads them to a list of all methods supported by `string` objects, and they find the one they need.

26

5

Students A and B are paired. A chats with their classmate C about the homework, and together A and C come up with some good strategies for designing a solution, strategies which A remembers and mentions later on when A and B are doing their pair programming.

27

6

Students A and B are paired. They sit together in front of one computer. A starts working on the CS 70 assignment. B pulls out a paper copy of a history paper and starts penciling in edits, while occasionally glancing up and making comments on A's code.

28

7

Students A and B are paired. Because they work on different campuses, they work on separate computers in their own dorm rooms using "screen sharing" and on-line chat to discuss and edit the same file at the same time.

29

8

Students A and B are paired. Before they get very far, B falls ill. Several days later, just before the assignment is due, the professors are asked for an extension (because B was too sick all week to work).

9

Students A and B are paired. They have a bug in their code they just can't figure out. In a public post on Piazza, they paste the lines of C++ code that they think are responsible and ask for help.