Lecture 1a: Welcome!

CS 70: Data Structures and Program Development

Tuesday, January 22, 2019

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

Course Resources

www.cs.hmc.edu/cs70

Pair Programming

Question

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

Question

How can “Pair Programming” skills improve with practice?

The Rock Lab (BK B111)

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.

Class Exercise: Scenarios

1. Take scenario handouts.
2. In groups of 4, decide how each of your scenarios is best described:
    - Encouraged
    - Acceptable
    - Discouraged
    - Forbidden

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.

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.

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.

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.

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.

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.

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.

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.

10

Students A and B are paired. Their Binary Search function goes into an infinite loop on some inputs. They can trace through the code and see exactly where and why it loops; they just aren't sure how to fix the algorithm (without breaking other cases). After some discussion that doesn't go anywhere, they decide to take a break and come back the next day with rested minds and fresh eyes.

Programs and Memory

How does a program run?

In a groups of 4, what do you remember about HMMM?

Compiling

Source Code

fourtwo.cpp:

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

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)
```

Object Code

fourtwo.o:

```
^?ELF^B^A^A^@^@^@^@^@^@^@^@^A^@>^@^A^@^@^@^@^@^@^@
^@^@^@^@^@^@^@^@^@^@\230^A^@^@^@^@^@^@^@^@^@^@^@^
@^@^@@^@^H^@^A^@UH\211å1ÀÇEü^^^@^@^@ÇEø^L^@^@^@\213Mü^
CMø\211Mô]Ã^@clang version 6.0.0-1ubuntu2 (tags/RELEAS
E_600/final)^@^@^@^T^@^@^@^@^@^@^@^@^@AzR^@^Ax^P^A^[L^G^H
\220^A^@^@^\^@^@^@^\^@^@^@^@^@^@^@^@^_^@^@^@^@A^N^P\206^BC
```

(line breaks added and more content omitted for space. . .)

Executable

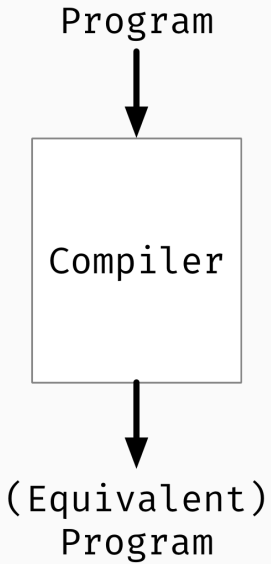
`fourtwo`:

```
^?^B^A^A^@^@^@^@^@^@^@^@^@^B^@>^@^A^@^@^@\260^C@^@^@^@^@
^@@^@^@^@^@^@^@^@\220\260^X^@^@^@^@^@^@^@^@^@^@@^@8^@
^@@^@^Z^@^Y^@^F^@^@^@^D^@^@^@@^@^@^@^@^@^@^@@^@@^@^@^@^@
^@@^@@^@^@^@^@^@\370^A^@^@^@^@^@^@\370^A^@^@^@^@^@^@^H^@
^@^@^@^@^@^@^C^@^@^@^D^@^@^@8^B^@^@^@^@^@^@8^B@^@^@^@^@^
@8^B@^@^@^@^@^@^\^@^@^@^@^@^@^@^\^@^@^@^@^@^@^@^A^@^@^@^@
```

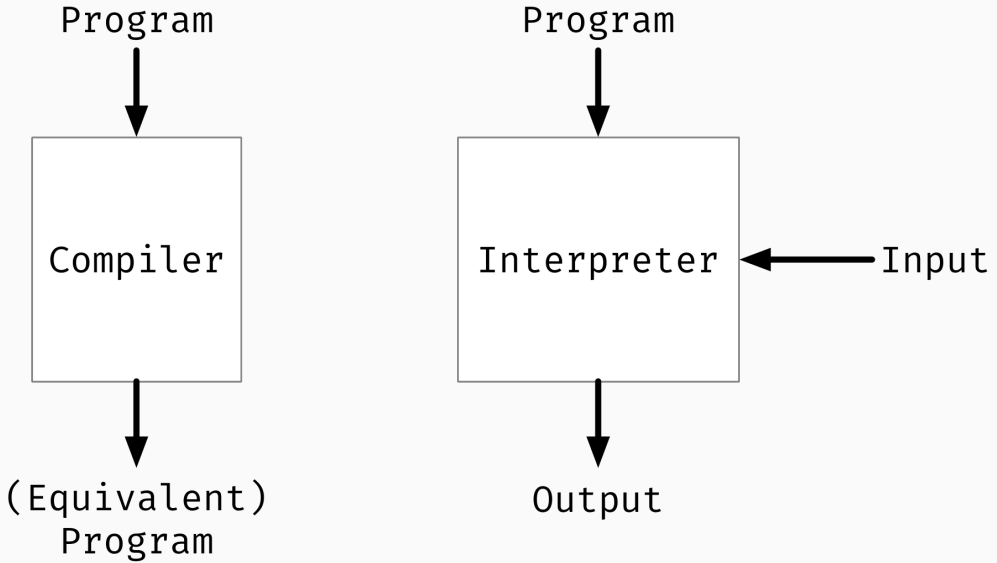
(line breaks added and content omitted for space. . .)

Compiling vs. Interpreting

Compilation



Compiling vs. Interpreting



Compiler or Interpreter?

1. An *Assembler* for HMMM.
2. The "Run" button in Eclipse (when you have a Java program loaded).
3. The CPU in a laptop computer.

Running start

- Looking for partner? Post on Piazza or pair up in Lab on Friday.
- Homework 00: availabile now, due Friday – getting started with CS70 tools.
- Homework 01: available Thursday, due next Wednesday – getting started with C++.