Lecture 30: Introduction to C

History of C
- Developed by Dennis Ritchie between 1969 and 1973 at AT&T Bell Labs
- Used to re-implement the Unix operating system
- One of the most widely-used languages of all time

Travel
- Prof. Kim will be gone this Thursday and Friday
  - No office hours during that time
- Prof. Mawhorter will be gone next Wednesday, Thursday, and Friday
  - Prof. Kim will cover lab next Wednesday

History of C
- Constructs map efficiently to machine instructions
- Used in same places as assembly language:
  - Operating systems
  - Supercomputers
  - Embedded systems (e.g., your microwave)

For more see https://en.wikipedia.org/wiki/C_(programming_language)
Learning C

- Mostly a subset of Java
- Debugging is difficult (give yourself time)
- A whole new language of compile errors
- Resources:
  - Reference: http://en.cppreference.com/w/c
  - Tutorial with web IDE: http://www.geeksforgeeks.org/c/

High-Level Differences

Java:
- Compiles to byte code;
- runs on the JVM
- Garbage collection
- High-level concepts: objects, exceptions
- Safety by default

C:
- Compiles directly to CPU instructions
- Garbage (no collection)
- No high level concepts;
- close to assembly
- Safety? What’s that?

Similarities

- Primitives: int, float, double, boolean, char, “void”
  - C distinguishes between signed and unsigned.
- Syntax
  - Curly brackets and semicolons
  - Function and variable declaration
  - Control constructs: while, for, if-else, switch
  - The “.” operator for instance variables

C Structs vs Java Objects

C:

```c
struct point {
    int x;
    int y;
};
```

Java:

```java
public class point {
    public int x;
    public int y;
};
```

- Can only have public instance variables
- No methods
- Can be stack allocated like primitive types
- Cannot directly contain itself
How to Program in C?

Text editor and terminal commands!

- Editor:
  - Anything with syntax highlighting
  - Emacs and Vim are classics
  - Aquamacs, Atom, Sublime

- Compiler
  - clang on Macs, clang or gcc on little.cs.pomona.edu

Makefiles

- Executes shell commands to build stuff
- Stores build configuration like Eclipse did for Java
- Commands make, make run, make memcheck, make package, make package-nocompile, make clean

Example Code

- hello.c
- vector.c