Introduction to C:
Programming without Class

CS 62 - Spring 2016
Michael Bannister

Some text from “Essential C” by Nick Parlante

History of C

The C language was originally developed by Dennis Ritchie between 1969 and 1973 at AT&T Bell Labs, and was used to re-implement the Unix operating system. It has since become one of the most widely used programming languages of all time!

History of C

The C language provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as software for computers ranging from supercomputers to embedded systems.

For more on the history see: https://en.wikipedia.org/wiki/C_(programming_language)

Learning C

- C is missing object oriented features of Java, but most other parts are the same.
- Finding bugs in C code is very difficult !!! Must leave a lot of time for debugging.
- Compiler warning/error message are getting better, but still not as good as Java
- There are many helpful resources to learn C online
  - Complete reference with examples: http://en.cppreference.com/w/c
  - Nice tutorial with web IDE: http://www.geeksforgeeks.org/c/
  - An interesting book by Zed Shaw: http://c.learncodethehardway.org/book/

Official Texts

- **Essential C** by Nick Parlante
- **Pointers and Memory** by Nick Parlante
- **A Rapid Introduction to C** by Michael J Bannister
  A modernization of the above two texts.

Differences with Java

Java

- Compile once run anywhere: byte code, JVM
- Garbage collection
- High-level concepts: objects, exceptions, etc…
- Safety by default

C

- Compiles to "native" object code; runs "bare metal"
- No garbage collection
- No high level concepts: close to assembly
- Safety left to you and some third party solutions

Similarities with Java

- **Primitive types:** short, int, long, float, double boolean, char, "void"
  C adds a distinction between signed and unsigned integer types!
- Same syntax for:
  - Curly braces and semicolons
  - Functions
  - while, for, if-else, switch
  - The "." operator for instance variables

C Structs vs Java Objects

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

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

- Can only have public instance variables
- No methods at all!
- Can be stack allocated like a primitive type
- Cannot directly contain an instance variable of itself: No recursive structs!

CS 62 C Environment

Text editor and terminal!

- Text editor: Any C aware text editor will work, but I recommend atom.
- Compiler: Clang 3.8 on OS X El Capitan. Note this is not the default version of clang on OS X!
- Makefile: types out long terminal commands so you don't have to! Examples of using out official make file will be given on Monday.

Example Code

- Some sums
- Points and dots