

# System-level Programming

Avery Karlin

Fall 2015

# Contents

<b>1</b>	<b>Learning C</b>	<b>3</b>
1.1	C Primitive Variable Types . . . . .	3
1.2	C Programming . . . . .	3

# 1 Learning C

## 1.1 C Primitive Variable Types

1. All C primitives are numeric, divided purely based on variable size, and integer or floating point
  - (a) C variables have sizes based on the platform they were compiled by and for, such that `sizeof(type)` can be used to determine the size in bytes
  - (b) On a standard computer, `int = 4`, `short = 2`, `long = 8`, `float = 4`, `double = 8`, and `char = 1` bytes (8 bits to a byte)
  - (c) Types can also be specified as unsigned, such that it is not able to be given a negative value
2. Boolean values are numbers, such that 0 is false, and all nonzero numbers are considered true
3. Character literals can be represented inside single quotes rather than use a number, and Strings, though not an object, can use a double quotes literal
  - (a) Strings are created by character arrays, using a null character (value 0), to show the end of the array, allowing it to be modified easier
4. Variables are able to be initialized within a for loop, but are not able to be declared, such that it must be before the loop

## 1.2 C Programming

1. All C programs are made up of a series of functions, run within the main function, which returns an integer (typically 0, or other values for errors)
  - (a) They are compiled through “`gcc file.c -o program_name`”, then run through “`./program_name`”
2. Libraries are added, either `.h` files from the current directory through `#include “file.h”` or through premade libraries by `#include <file.h>`
  - (a) All files typically start with calling the C library with `#include<stdio.c>` and `<stdlib.h>`

## 1.3 C Structures

1. “`printf(text, var1, var2)`” is used to print a String in terminal, where the text is a formatted string, with placeholders for variables following
  - (a) `%f` is a placeholder for a float, `%d` for double, `%c` for char, `%s` for string, `%f` for pointer, `%lf` for double, `%ld` for long, and `%d` for int