C Reference Sheet (Author: Jack L. Watkin)

Compiling

gcc	GNU C Compiler (ex. `gcc file.c`)
-c	Generate object files (.o files)
-o	Specify name for output executable

Preprocessor Macros

General	C Preprocessor Directives are directives processed at compile time that change the text of a program
`#define`	Create a symbolic constant or expression
Examples:	`#define NOR(A,B) !(A || B)` `#define TRUE 1`

Storage and Linkage Classes

Static Var, outside function	Storage: static Linkage: internal
Static Var, inside function	Storage: static Linkage: none
Non-static Var, outside function	Storage: static Linkage: external
Non-static Var, inside function	Storage: automatic Linkage: none
Static function	Storage: none Linkage: internal
Non-static function	Storage: none Linkage: external

Strings

Strings are character arrays that are terminated with the `NULL` character(literal 0).

Strings must be declared large enough to hold the string plus `NULL` termination (A string to hold `"Hello"` would need to be 6 characters long)

Double quotes include `NULL` character (`"Hi"` -> Hi plus `NULL` termination.)

String functions and manipulation

`snprintf`	Formatted print to string, protecting against buffer overflow
`sscanf`	Formatted read from string
`strlen`	Returns length of string without null termination
Examples:	`snprintf(d,10,"%s",s);` -Copies no more than 10 chars from `s` to `d`.

Standard and File I/O

`printf`	Formatted print to `stdout`
`fprintf`	Formatted print to file
`scanf`	Formatted read from stdin
`fscanf`	Formatted read from file

Examples:	`printf("%s\n", buffer);`
	-prints buffer with newline
	`scanf("%d %d", &i, &j);`
	-Read two ints from stdin into i and j

Opening File(s)

`fopen`	Opens a file
`fclose`	Closes a file
r	Opens for reading (if the file exists)
w	Creates a new file with the specified file name (overwriting existing files of the same name)
a	Opens for appending If the file does not exist, creates the file.
r+ w+ a+	Opens for reading and appending (if the file exists) Otherwise, creates file

Examples:	`fp = fopen("f.c","w+");`
	-Opens f.txt for reading and appending
	`fclose(fp);`
	-Closes the file *fp*

String Format Specifiers

%s	String of characters (null terminated)
%[*n*]s	String of *n* characters
%c	Single character
%d or %i	Signed integer
%u	Signed integer
%f	Floating point
%x	Unsigned hexadecimal integer
%p	Pointer address

Data Types

`char`	Character (1 byte)
`short`	Signed short integer (2 bytes)
`int`	Signed integer (4 bytes)
`long`	Signed long integer (8 bytes)
`float`	Single-precision float
`double`	Double-precision float
`stdint.h`	Contains definitions for fixed size integers (`int8_t`, `int16_t`, `int32_t`, `uint8_t`, `uint16_t`, `uint32_t`)
`signed` `unsigned`	Modifier that makes a variable signed or unsigned

Dynamic Memory

General	Dynamic memory is that must be explicitly allocated at runtime and freed when it is no longer needed.
`malloc`	Allocates dynamic memory returning a pointer to the allocated memory.
`free`	Function for freeing allocated dynamic memory.

Examples:	`Arr = malloc(5);`
	-Allocates an array of 5 bytes
	`free(Arr);`
	-Frees allocated memory