10/1/21 Notes

Word of Caution

- function C are not the same in math function
  - they may/not return a value (void returns nothing)
  - may/may not have side-effects
- subroutine in other languages
- In Java, it's a method

How is it different?

- math uses domain and range

- In C, we call function
  - function it returns value

function in programming:

- block of code that performs a certain task.
- defined exactly once.
- must be declared before they are used.
- can declare & call a function as many times as desired.
- main()
  - is a special function.
  - It run when program starts.
  - All other functions are subordinate main()

Why do we like functions?

- functions should:
  - define abstractions
  - give names to segs. of code.
  - hide the implementation
- use them to:
  - refactor repeated segs. of code.
  - simplify code to aid understanding
- Functions should never be:
  - Arbitrary segs. of statements

Function definition.
- function head
  - return type
    - defines type of functions return value
    - return-type could be void or any obj. type (except. array)
  - function_name()
  - parameters
    - contained in comma separated list at decl.
    - if function has no parameters then void.
- Function block/body
  - declarations
    - declared variables inside a funct. bdy are local
  - assign new

Return values.
- funct. returns a value
- may be void
- return a struct (not recommended)
- return a pointer
  - not an array

Function naming.
- same naming rules as variable
- can't start
  - start w/ a number or any punctuation other than _ or &
- use snake case
  - my_function_name

Parameters or Arguments
- If we have f(x) = x log(x+1) and we write f(a) we substitute a for x and get a log(a+1).
  - this is call-by-name and is rare.
  - C supports this as textual subs is process in C Preprocessor.
- most prog. lang, use either call-by-name, call-by-reference, or both.
- C uses call-by-value, except for arrays, and only b/c of their relation to pointers.

Parameters
- formal parameters.
  - name of parameter as it's used in body
- Actual parameters,
  - name of value that is passed to func
  - the value can be copied to the formal param.
  - or a ptr/ reference to the actual parameter
- Copy-in-Copy out means that inside. to be copied, value is copied back out
  - C does not support this.

Call-by-value.
- all functions use this.
- Arguments passed into a function are copied.
  - any changes made to the parameters inside the function has no effect.
- the called function copies the values i

Call-by-reference
- refs.

Ternary operator
```
h=(x<1) ? 1 : x;
if (x<1) {
return 1;
} else {
return x;
}
```

swap()
- does not have true call-by-reference so we use pointers.
- Addresses instead of values are passed to arguments

```
void swap (int *a, *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
    return;
}
int void main(void){
    int x=5;
    int y=7;
    swap(&x, &y);
    printf("val x = %d", x);
    printf("val y = %d", y);
```

function prototype
- syntax:

return_type function_name(param);

- prototype must be declared either at beginning of program or in included header file.

#include
- a preprocessor directive,
- Before compiling, C source files are processed by preprocessor
- preprocessing is a macro processor to transform programs before compiling
- macros in C operate through text replacement
- tells compiler of your file into current file.
- used to include functions defined in other libraries

#define
- A preprocessor directive that defines a macro for the program
- C preprocessor performs all text replacements for defined macros prior to compilation

Conditional directives
- a set of preprocessor directives that help Cond. statements to include code selectively.

- #ifndef - execute statements when NAME is defined

Header file
should only contain things that are shared b/t source files

#ifndef vs. # pragma once.
· #ifndef checks macro will always work (use it more portable)

extern
· extends visibility of var and functions such that they can be called by any program file.

Static
· can be declared in and outside function.