CSCI E-50 WEEK 2

TERESA LEE

FEBRUARY 5, 2018

TODAY

- Introductions
- Resources Review
- Debugging
- Arrays
- Functions and variable scope
- Command-line arguments
- Problem Set 2 Preview

ABOUT ME

HBSc PSYCHOLOGY & HUMAN BIOLOGY, U OF TORONTO (2009)

MDes INCLUSIVE DESIGN, OCAD U (2018)

Email: TERESA@CS50.NET

Sections: Mondays 6-8 pm EST

Office Hours: Thursdays 9-11 pm EST

ABOUT YOU

Share:

- YOUR NAME
- WHERE YOU LIVE
- WHAT WOULD YOU LIKE TO GET OUT CS50?
- PREVIOUS CS EXPERIENCE?

MY JOB AS A TF

- SECTIONS
- OFFICE HOURS
- E-MAILS: 12-24 HOURS RESPONSE WINDOW
- GRADING

WHAT TO EXPECT FROM SECTION

- Review of core concepts
- Hands-on practice problems and activities
- Pset questions
- Q & A
- Anything and everything in between

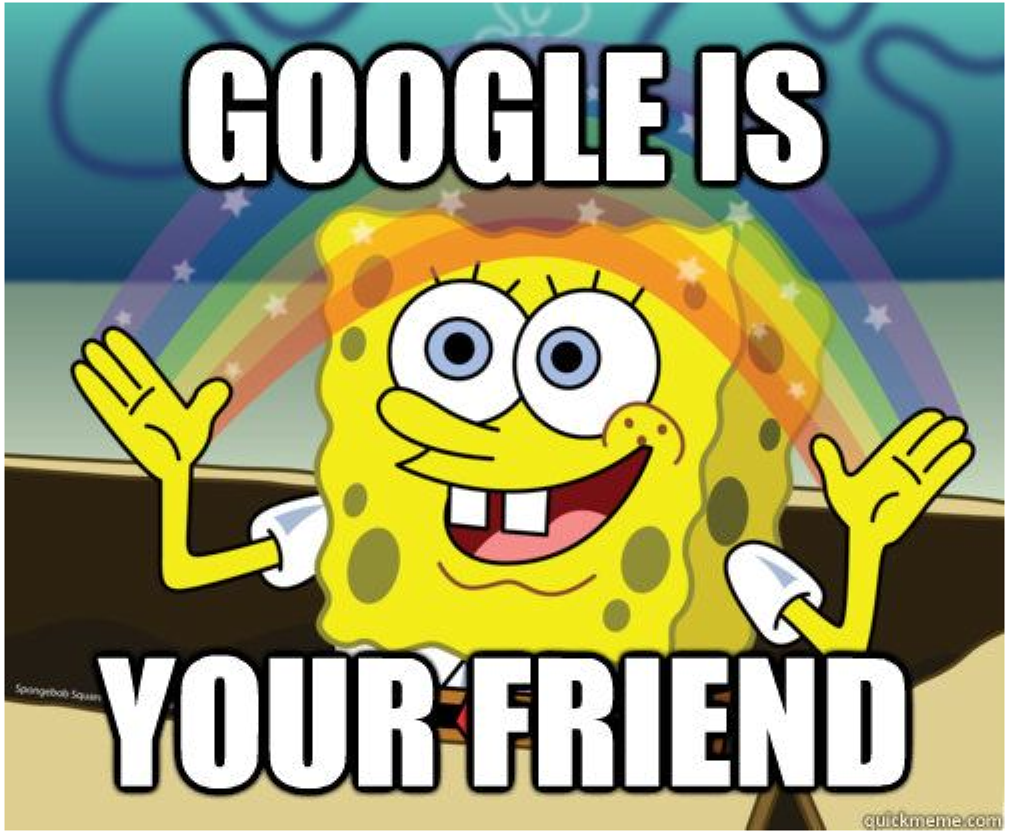
GROUND RULES

- Come prepared
  - Watch lectures
  - Read pset specification
  - Prepare questions
- Share feedback
- Have fun!

RESOURCES

- Lecture videos
- Scribe notes
- Reference sheets
- Walkthroughs
- Shorts
- Reference50
- Style Guide
- Your Peers
- CS50 Discourse
- Office hours Everyday except Sunday & Monday

* Info on setting up CS50 IDE Offline


GOOGLE IS
YOUR FRIEND
Spongebob Squar
quickmeme.com

QUESTIONS?

DEBUGGING

- `eprintf`
- `debug50`
- `help50`

eprintf

- eprintf with context.
- Tells you exactly what line of code in their program triggered the eprintf call as well.
- Ex 1

debug50

- The graphical debugger for chasing down bugs in code.
- You can:
  - step into
  - step over
  - display variables' values
  - change variables' values
- Ex 2

help50

- can prepend calls to, e.g., make or clang to help explain the sometimes confusing error messages that can result.
- Ex 3

[redacted]

ARRAY

To initialize an array

```
int score[0] = 0; // zero index all
arrays!
int score[1] = 1;
int score[2] = 2;

                 or

int score[] = {0, 1, 2}; // size
based on the number of entries
```

```
// make an array

<datatype> <name>[<size>];

    char alpha[26];

    Int score[5];

// iterate over the array’s members
```

ARRAY

```
int bar[5] = {0,2,3,4,5}

int foo[5]

Bar[0]

Bar[1]



Can you do foo = bar??
```

Let’s Look at Examples

Ex 4

Ex 5

STRING

Just an array of characters!

Final index of a string in C is the null terminator ‘\0’, which tells a string that the string is over.

```
// declare string

String s = “teresa”;

// what happens when I index into
s[i]?

Printf(“%c\n”, s[0]);

Printf(“%c\n”, s[1]);

Printf(“%c\n”, s[6]); \0

Printf(“%c\n”, s[7]);
```

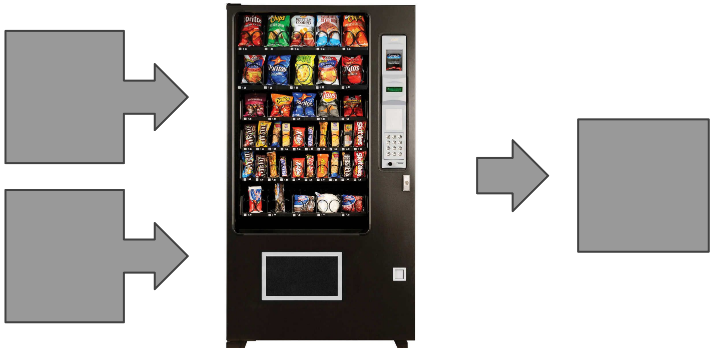
GANDIA
GANDIA
GANDIA
GANDIA

FUNCTIONS

(1) take something in [parameters],

(2) do something [body], and finally

(3) spit out an answer [return value].



Why use a function?

- Simplification
- Organization
- Reusability

Creating a function

1. Declaration
2. Definition
3. Function calls

```
<return-type> <name>(<paremeter-list>);
```

FUNCTIONS

(1) take something in [parameters],

(2) do something [body], and finally

(3) spit out an answer [return value].

Why use a function?

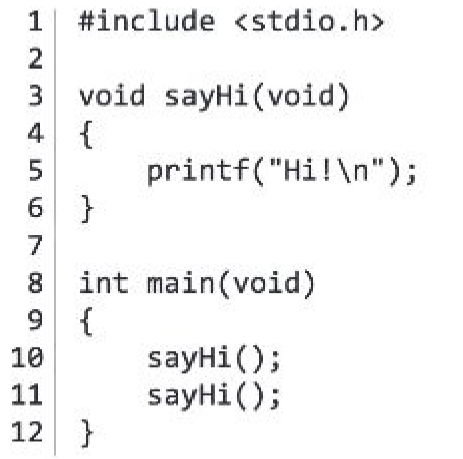
- Simplification
- Organization
- Reusability

Creating a function

1. Declaration
2. Definition
3. Function calls

```
<return-type> <name>(<paremeter-list>);
```

Let’s Look at Examples



```c
#include <stdio.h>

void sayHi(void)
{
    printf("Hi!\n");
}

int main(void)
{
    sayHi();
    sayHi();
}
```

- Let’s look back at Ex 2!
- Ex7

SCOPE

Scope is a characteristic of a variable that defines from which functions that variable may be accessed.

- Local variables: can only be accessed within the functions in which they are created { }.

- Global variables: can be accessed by any function in the program.

COMMAND LINE ARGUMENT

Allows you to pass arguments into the main function of the program by specifying the arguments at the command line.

Benefit? Offers an alternative means of providing input to a program beyond just requesting input while program is running.

argc - argument count

argv - argument vector(array of chars or string)

```
int main(void)
```

Becomes…

```
int main(int <argc>, string <argv>)

//this lets the program know that it needs to expect and process command line parameters.
```

Where have you already seen them?

argc

command	argc
./greedy	1
./greedy 10 cs50	3
./cube 3 5 7	4

argv

```
command: ./greedy 10 cs50
```

argv indices	argv contents
Printf (“%s\n”, argv[0])	greedy
argv[1]	10
argv[2]	cs50
argv[3]	nothing!

Let’s Look at Examples

Ex 6: Revamp ex 2! Require a user to enter input at the command line.

Pset2: Crypto

Choose two adventures:

- Implement Caesar’s cipher (less comfy)
- Implement Vigenère’s cipher (less comfy)
- Crack passwords (more comfy)

You’re welcome but not expected to implement all three;
if you submit all three, we’ll grade your two best.

Final words on pset2

- Remember they need to preserve case!
- All command line arguments are strings. How might they convert "1" in the above example to an int?
- <ctype.h>
- Be careful about rotating too far - use mod (%)!
- Vigenere is similar to Caesar, except in place of an int, a second string is used to encrypt the target word.
  - For example, entering "Doug Lloyd" and "hi" will output "Kwbo Stvgk".
- Pseudocode
- Don’t forget to comment!