INTRO TO PROGRAMMING

Bash Review, Compilation,
and Procedural Programming

REVIEW: SETUP AND Q&A

Windows Host

```
cd Desktop\vagrant-workspace
vagrant up
```

Apple Host

```
cd vagrant-workspace
vagrant up
```

Get the script to automate VM setup and assignment checking

```
# install if needed
sudo apt-get install wget

wget https://raw.githubusercontent.com/pappasam/npd/C1/c1_intro_programming/scripts/course_set
bash course_setup.sh
```

KEEPING IN SYNC

From now on you can run the script to ensure you have latest version of course repo and check homework completion with this command:

```
~/npd/c1_intro_programming/scripts/course_setup.sh
```

C LANGUAGE

```c
#include <stdio.h>

/* Write "hello, world\n" to stdout. Return 0.

int main(void)
{
    printf("hello, world\n");
    return 0;
}
```

Start editing a file to add the above contents by entering

`vim ~/intro-programming/assignment_4/hello.c` into the prompt.

THE MAKE TOOL

A tool for specifying build dependencies
and the commands to satisfy them

```
CC = gcc

hello: hello.c
	$(CC) -o hello hello.c
```

Start editing a file to add the above contents by entering

`vim ~/intro-programming/assignment_4/Makefile` into the prompt.

COMPILATION

```
# don't run mkdir if assignment_4 already exists
mkdir ~/intro-programming/assignment_4

cd ~/intro-programming/assignment_4
cp ~/npd/c1_intro_programming/scripts/a4_Makefile Makefile
sudo apt-get install make

make
```

C CHARACTERISTICS

- Imperative - define a sequence of instructions, as opposed to the result you want
- Lexical (Static) Scope - can only use a variable in block it's defined
- Static Typing - values can be of one 'type' such as character string or integer

C EXAMPLE: LOOPS

```c
#include <stdio.h>

int main(void)
{
    for (int i = 1; i <= 10; i++) {
        printf("Loop iteration: %d\n", i);
    }
    return 0;
}
```

BASH EXAMPLE: LOOPS

```
#/bin/bash
for i in $(seq 10); do
    echo "$i"
done
```

PYTHON EXAMPLE: LOOPS

```
# run with `python loop.py`
for i in range(10):
    print(i)
```

PROCEDURAL PYTHON

Procedural refers to the ability to reuse blocks of code called procedures or more commonly functions (not to be confused with "Functional Programming")

RECREATE TASK SAVER

- https://docs.python.org/3/library/os.html#module-os
- https://docs.python.org/3/library/functions.html#open
- https://docs.python.org/3/library/io.html#module-io

Examples and description in assignment_4 file on course repo:

https://github.com/pappasam/npd/blob/C1/c1_intro_programming/assignments/a4_file_system_scripting_challenges.md

NEXT TIME:

More Python and introduce coding style