Lecture 1

1 Basic Unix Commands

That is, how to use the black box called terminal.

```
ls
pwd
ls -a # (come back to . and ..)
ls -l
ls -al
mkdir CPE101
ls # (with flags?)
cd CPE101
ls
pwd
mkdir Lecture01
cd Lecture01
ls
pwd
cd .. # (oh!  magic!, explain ..)
cd ~
cd /Users/btjones/CPE101/Lecture01
cp ~/Files/smallfile . # (explain .)
ls -l
cat smallfile
cp ~/Files/largefile .
ls -l
cat largefile
less largefile # (q to quit)
rm smallfile # (be VERY careful with this!)
cp ~/Files/smallfile .
mv smallfile ..
cd ..
ls
mv smallfile ..
cd ..
ls
mv smallfile Lecture01
cd Lecture01
mv smallfile newfile # (can use mv to rename)
man mv # (forget how to use mv?  q to quit)
```

2 Python

We're using Python 3 vs 2, there are some minor differences that we won't talk about.
Run interpretor, we can use it as a calculator!

```
2 + 2
3 * 4
10 - 26
7 / 2
7 // 2
x = 5 # variables!  math!
x * 3
y = 2 * x + 4
y
greeting = "Hello" # we can store text!  we call these strings!
x > 2
y < 4
```

Variables, assigning values to them, show as a box. Use variable names that mean something! Some words cannot be used.

3 Operations

Python has a bunch of operators (+, -, *, /, %, //, **).
What is the value of `3 + 2 * 4`
What about `3 + 2       * 4`?
`(3 + 2)*4`?

3.1 Order of Operation

- Parentheses
- ** (right -> left)
- unary – (like `-5`)
- *, /, //, % (left -> right)
- +, - (left -> right)

3.2 Evaluating Expressions

Have them guess:

- `-3**2+7.0//2 # = -6.0`
- `3 + 6 // 4 - 2 ** 2 + 8 / 2.0 / (2 + 1) - 25 % 4 # = 0.3333333333`
- `8 / 3 # = 2.6666666666666665`

4 Editing Files

We can also put code in a file! Python files end with the extension `.py`.

```
cp ~/Files/code.py .
cat code.py
python3 code.py
```

5 Testing

```
cp ~/Files/testing.py .
```

Talk about unit testing! Talk about floats sucking!