Homework Policy

- Submit through NYU classes. Your responsibility to confirm it was uploaded correctly.

- We will deduct 10% for every day it is late. Will not be accepted after 7 days.

- If you have what you believe to be extenuating circumstances, explain them to me and tell me when you expect to be able to complete it. If I grant the extension I will state by what date you must turn it in.

- Do everything you can to stay on top of the homework. Everything builds on previous concepts in this course, so the more you fall behind, the harder it is to catch up.
Reading Ahead

- If you are inspired to read ahead and use techniques we have not yet covered in your programming assignments, that's ok…

- …unless I explicitly request you use a specific technique or construct.

- … and you must convince me in your comments that you understand what the code you use does, and why it's better than techniques we have already discussed.
Google Alert

- Please don't try to use Google (or Stack Overflow) to solve any programming assignments for this class.

- None of the assignments will require code techniques that haven't been covered in the book, the modules, and the lectures.

- To learn programming well, you have to start with the basics. Also, your exams will cover what we teach in class.

- Many online answers will use idioms that obscure what the code is doing, and won't help you learn, or may in fact do the wrong thing.

- Also, many online answers will be for earlier versions of Python. Python changed significantly between version 2 and version 3.
Test Your Code!

- Make sure your code runs before you submit it! You will get 0 points for non-working code.

- If your code will crash on invalid inputs, put a comment explaining that. Any other crashes that prevent your code from generating the correct output will cause you to get 0 points for that problem.

- Make sure to test with the input values given as examples in the problem description. You should also test with other values to make sure the code behaves as you expect. (Try to anticipate potential problem areas or "edge cases".)
Why Not Do This?

```python
bill_amt = float(input("Enter bill total: "))
tip = float(input("What percentage tip: "))
bill = float(format(bill_amt, ".2f"))
tip_amt = bill * (tip/100)
```
Why Not Do This?

```python
bill_amt = float(input("Enter bill total: "))
tip = float(input("What percentage tip: "))
bill = float(format(bill_amt,'.2f'))
tip_amt = bill * (tip/100)
```

- Inefficient to keep converting types
- Introduces greater rounding errors (every time you round, you introduce error up to $\frac{1}{2}$ of the precision you round to.)
- Format usually not needed until you're ready to print
Decision Structures Review
Trace the Output

```python
a = 5
b = 10

if a < b:
    print ("one")
if a > b:
    print ("two")
if a*2 == b:
    print ("three")
if b < a:
    print ("four")
```
Trace the Output

```python
a = 5
b = 10

if a < b:
    print ("one")
if a > b:
    print ("two")
if a*2 == b:
    print ("three")
if b < a:
    print ("four")
else:
    print ("five")
```
Trace the Output

a = 5
b = 10

if a < b:
    print ("one")
elif a > b:
    print ("two")
elif a*2 == b:
    print ("three")
elif b < a:
    print ("four")
else:
    print ("five")
Boolean Variables

... and Flags
Boolean Data Type

- The data type is called "bool"
- Possible values are True or False
- True and False are boolean literals
- Boolean values an be assigned to variables:
  
  a = True
  
  b = (x < y)
- One use for this is to record the value of a condition at one point in the program so that it can be checked and acted-on later. This is called a flag.
Using Modules
Modules

- Modules are containers for additional functions and code.

- Generally, modules are used to wrap related functionality together, and to make it available optionally – so you don't load it (and take up memory) if you don't need it.

- Modules are also called Libraries and Packages.

- Python has many modules that provide standardized solutions for many problems that occur in everyday or specialized programming.

- Modules we've seen so far:
  - turtle
  - math
  - sys
Using Modules

- You "load" a module (tell Python you are going to use it) with the import statement
  
  ```python
  import turtle
  import math
  ```

- `import` looks for a file with that name and the .py extension. First it looks in the "local" directory (the same directory where your program file lives). If it can't find a file with that name there, it looks in the global modules directory for your Python installation.

- So if you're trying to use the turtle module, don't have a program file named "turtle.py" in your local directory
Using Modules

- Once your module is loaded, you use functions or values from it using the "dot" syntax:
  
  ```python
  turtle.penup()  # function call
  math.sqrt(n)   # function call
  sys.float_info  # a constant
  ```

- If you try to use anything from a module you haven't imported, Python will tell you it doesn't know what you're talking about.

  ```python
  >>> math.sqrt(4)
  Traceback (most recent call last):
    File "<pyshell#4>", line 1, in <module>
    math.sqrt(4)
  NameError: name 'math' is not defined
  ```
By The Way

- Modules are regular python files!
- Which means you can make your own modules (eventually)
- In Idle, when you run your file, you use the menu command "Run Module"
- Also, if you want to know how a function in a module works, you can (usually) just open the source code and examine it!
Generating Random Numbers
Generating a random integer

- Sometimes you want your program to do something unpredictable

- You can ask your programming language to generate a “random number” – from there you can use this number to construct a somewhat random set of running conditions

- The "random" module contains functions for generating random values

- `randint()` takes two parameters (a starting integer and an ending integer) and returns a random integer in this range
Random Integer Example

# ask Python to import the random module
import random

# generate a random number
num = random.randint(1,5)

print("your lucky number is", num)
Programming Challenge: Rock, Paper, Scissors

- Write a program to ask the user to select one of three options - Rock (r), Paper (p) or Scissors (s)
- Use the random.randint() function to select an option for the computer
- Determine the winner and print the result.
  - Rock beats Scissors
  - Scissors beats Paper
  - Paper beats Rock