POLYTECHNIC COMPUTER SCIENCE - SPRING 2013
PROGRAMMING PROJECT #1
COIN FLIPPER

ASSIGNMENT OVERVIEW
In this assignment you'll be creating a program called *coinflipper.py*, which will simulate flipping a coin a specified number of times.

This assignment is worth 50 points and is due on the crashwhite.polytechnic.org server at 23:59:59 on the date given in class.

BACKGROUND
Simulations are an important use of computer programming. Simulations allow a computer to model all sorts of systems, from the very simple to the very complex, and provide a means for studying the behaviors of those systems. The simulation in this exercise is a relatively simply one, but allows us to collect an enormous amount of data, far faster than we could if we had to physically flip a coin.

PROGRAM SPECIFICATIONS
Create a Python program that:

a. asks the user his/her name, and uses that name to address the user in the program
b. asks the user if her or she would like to simulate flipping a coin to see how often it comes up "heads" or "tails"
c. if the user *does* want to perform the simulation, asks how many simulated coin flips they want to perform
d. simulates flipping the coin the indicated number of times, and print out the results, along with some statistics
e. asks the user if he or she would like to perform the simulation again. The program should continue performing the simulation until the user indicates s/he is done.
f. has self-documenting identifiers and comments throughout.

DELIVERABLES
coinflipper-lastnamefirstinitial.py
(The instructor, for example, would submit the file "coinflipper-whiter.py" .)
You should keep a working copy of this file in your home folder on the server, and a backup copy of the file elsewhere. To submit your assignment for grading, copy your file to the directory /home/whiter/Public/Dropbox/coinflipper at *crashwhite.polytechnic.org* before the deadline.

1. Please be sure to use the specified file-naming convention.
2. Save a copy of your file on your hard drive, flash drive, etc..
3. Your grade will be based on the file you upload to the coinflipper folder.

ASSIGNMENT NOTES

To complete this assignment you'll need to know about two things that we haven't fully discussed yet.

1. A conditional *while-loop* will be necessary to find out whether or not the user wants to continue running repeated simulations. Consider using something like this:

```
done = False       # done is a Boolean variable
while (not done):  # while we haven't finished
    # Do the simulation in here
    # Lots of statements
    # in here
    # Now, check to see if they want to do it again.
    more = raw_input("Would you like another simulation (Y/N)?"
    if (more == "N"):
        done = True
print "I hope you enjoyed our simulation. Thanks!"
```

2. To simulate flipping the coin, you'll need to use a pseudorandom function, which happens to be available to us in a module called **random**.

```
>>> import random               # Gets the random module
>>> print random.random()       # Gets a pseudorandom decimal
0.223112306461
>>> print random.randint(0,4)   # Prints random integer from 0-4,
                                # inclusive
4
>>> print random.randint(0,4)
0
```

GETTING STARTED

1. With paper and pencil, and perhaps in collaboration with a partner, identify what the main components are that you'll need to include in your program.
2. Sketch out the basic flow of your program using a flowchart.
3. Write some pseudocode that you can use to begin implementing those main components.
4. Either on your personal computer or on the *crashwhite.polytechnic.org* server, open up two windows: a text editor to write the program (on the left), and a shell to perform test runs of your program on the right.
5. Use the text editor to begin writing your program, perhaps beginning with the pseudocode, and then filling in more details to various parts of the code.
6. Save your program with the required name.
7. Copy this initial version of the program to the *crashwhite.polytechnic.org* server (to make sure that the upload/copy process works).

```
$ scp einsteina-coinflipper.py
einsteina@crashwhite.polytechnic.org:/home/whiter/Public/Dropbox/co
inflipper  # a single command, all on one line
```

8. Switching to the shell, run your program as new features are added, making sure to fix old problems before adding new components. You'll repeatedly run through this edit-run, edit-run process to find bugs and fix them.

9. When your program is completed (but before the deadline), copy it to the server as indicated in #7 above. The newer version of your program will replace the old one there.

QUESTIONS FOR YOU TO CONSIDER (NOT HAND IN)

1. What is the difference between *random* and *pseudorandom*? How are pseudorandom numbers generated by a computer?
2. The number of heads and tails that come up in your simulations will very rarely be equal. Does this indicate a failure in the pseudorandom number function?
3. What happens to your program if the user enters numeric digits (say, 137) for their name? What happens if the user enters text (say, "eleven") for the number of coin flips they want to see?
4. What happens if the user enters 100.0 for the number of coin flips?
5. How much faster is the program then you when it comes to flipping coins? Determine how long it takes you to flip a coin 10 times, and calculate the percent difference between your time and the computers.

SAMPLE INTERACTIONS

Sample output from running your program might look something like this:

```
PhileasFogg:Desktop rwhite$ python coinflipper-whiter.py
Hi! What's your name? Richard

Well Richard, this program simulates coin flips.
How many times would you like to see a coin flipped? 1000

Of your 1000 tosses, there were 461 heads and 539 tails.
Would you like to do another simulation (Y/N)? Y
How many times would you like to see a coin flipped? -3
I need a number > 0 if I'm going to flip this coin!
How many times would you like to see a coin flipped? 50
Of your 50 tosses, there were 23 heads and 27 tails.
Would you like to do another simulation (Y/N)? N

PhileasFogg:Desktop rwhite$
```