

**Due:** Friday, 14 March 2014

**Collaboration:** If you collaborated with someone during lab to create a program, please acknowledge that person by adding their name to the individual program file. This is especially important if you worked with different people on different days. You must work on the last problem individually.

**Submission:** Turn in a printed copy of your work at the beginning of class. Please use a fixed width (monospace) font for your program text (e.g., Courier, Lucida Typewriter, etc.).

1. Submit program `swap.py` from the *Python* lab (Fri 2/21), Exercise 3 part 11.
2. Submit program `imagesize.py` from the *Python: Numbers* lab (Mon 2/24). This program was started in Exercise 2 part c, and improved in Exercise 3 part c. You should turn in the 3(c) version.
3. Submit program `favorite.py` from the *Python: Conditionals and Repetition* lab (Wed 2/26). This program was started in Exercise 1(b) and continued in 1(d).
4. Submit program `blastoff.py` from the *Python: Conditionals and Repetition* lab (Wed 2/26).
5. Submit program `decoder.py` from the *Python: Compressed ASCII Art* lab (Fri 2/28).
6. Sometimes we want program behavior to vary in useful, but unpredictable, ways. The first line of the following short program imports a subroutine that we can use to generate random (unpredictable) numbers. (This import need only be done once, making the subroutine available throughout the program.) We then call a subroutine to generate a random integer between 1 and 9, printing out the result. If you run this program, you would likely see a different result each time.

```
import random

num = random.randrange(1,10)
print "Here is a random digit: " + str(num)
```

Using this snippet as a starting point, write a program called `flashcards.py` that helps students learn their times tables by repeatedly generating multiplication problems (two random numbers between 1 and 9), asking them for the answer, and congratulating them if they are correct, while printing out the correct answer if they are not. The program should continue giving problems in an infinite loop until the user quits it (i.e., by typing *Control-C*).

Some things to keep in mind:

- You learned how to program an infinite loop in Exercise 2(a) of the *Python: Conditionals and Repetition* lab (Wed 2/26)
- Make sure you convert your data to the right types! Numbers should be multiplied by numbers, and strings should be concatenated with strings.
- While you asked the user to enter two numbers in Exercise 1(b) of the *Python: Numbers* lab, you can instead use the procedure `random.randrange(1,10)` twice to get two numbers.

The following might represent two example runs of your program.

```
What is 5 x 3? 13
```

```
Sorry, that is incorrect. The correct answer is 15.
```

```
What is 2 x 9? 18
```

```
Congratulations! That's correct.
```