Homework 1 (Revised)

Due: October 10, 2019 **Points:** 100

1. (*20 points*) The Konditorei coffee shop sells coffee at $10.50 a pound plus the cost of shipping. Each order ships for $0.86 per pound + $1.50 fixed cost for overhead. Write a program that calculates the cost of an order. Your program must prompt the user for the weight of the order in pounds.

   To turn in: Please turn in the program in the file *kondit.py*.
2. (*35 points*) Write a program to calculate the area A of a triangle given the length of its three sides a, b, and c using these formulae:

$$s = \frac{a+b+c}{2}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

   Prompt the user for the lengths of the three sides, one per line.

   To turn in: Please turn in the program in the file *tri.py*.

   Revision: The previous version had input entered as three numbers on the same line, separated by commas. This revision has the user enter one number per line.
3. (*45 points*) Write a program that approximates the value of π by summing the terms of this series:

$$\frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \ldots$$

   The program is to prompt the user for n, the number of terms to sum, and then output the sum of the first n terms of this series. (If n is 1, output the value of the first term only.) Your program is also to print the absolute value of the difference between the approximation and the value of `math.pi` to see how accurate the approximation is.

   To turn in: Please turn in the program in the file *pi.py*.

Extra Credit

E1. (*20 points*) Abraham Sharp developed an infinite sum that produces π:

$$\pi = \sum_{k=0}^{\infty} \frac{2(-1)^k 3^{\frac{1}{2}-k}}{2k+1}$$

   (a) (*15 points*) Compute and print the resulting approximations to π for the first 5, 10, ..., 50 terms. Each line of your output should look like this:

```
After 5 terms, the approximation is 3.1426047456630846
```

   Your number may differ; this is intended to show you the format of the output only.

   (b) (*5 points*) After what term does adding extra terms stop improving the approximation?

   To turn in: Please turn in the program in the file *sharp.py*. If you answer the second part, put the number in a comment, clearly labeled, at the top of the file.