Homework 3

Due Date: April 22, 2009 at 5:00PM **Points**: 100

Written Exercises

1. (*2 points each*) Show the result of evaluating each expression. Be sure that the value is in the proper form to indicate its type (int, long int, or float). If the expression is illegal, explain why. Please assume the math library has been imported via `import math`.
   a. `4.0 / 10.0 + 3.5 * 2`
   b. `10 % 4 + 6 / 2`
   c. `abs(4 – 20 / 3) ** 3`
   d. `math.sqrt(4.5 – 5.0) + 7 * 3`
   e. `3 * 10 / 3 + 10 % 3`
   f. `3L ** 3`
   [*text*, §3.8, Discussion problem 1, modified]
2. (*4 points each*) Show the output from the following fragments:
   a.
   ```
   for i in range(1, 11):
        print i * i
   ```
   b.
   ```
   for i in [1,3,5,7,9]:
        print i, “:”, i**3
   print i
   ```
   c.
   ```
   x = 2
   y = 10
   for j in range(0, y, x):
        print j,
        print x + y
   print "done"
   ```
   d.
   ```
   ans = 0
   for i in range(1, 11):
        ans = ans + i * i
        print i
   print ans
   ```
   [*text*, §3.8, Discussion problem 4]

Programming Exercises

Remember to turn in your error logs and refinement files.

3. (*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. Please turn in the program in the file `kondit.py`.
   [*text*, §3.8, Programming Exercises problem 5, modified]

4. (*20 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 separated by commas on a single input line. Please turn in the program in the file `tri.py`.
[*text*, §3.8, Programming Exercises problem 9, modified]

5. (*32 points*) Write a program that approximates the value of π by summing the terms of this series: 4/1 – 4/3 + 4/5 – 4/7 + 4/9 – 4/11 + … . 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. Please turn in your program in the file `pi.py`.
[*text*, §3.8, Programming Exercises problem 15, modified]

Extra Credit (Programming)

Remember to turn in your error logs and the refinement file.

6. (*20 points*) You have seen that the math library contains a function that computes the square root of a number. In this exercise, you are to write your own program for computing square roots. One way to solve this problem is to use a guess-and-check approach. You first guess what the square root might be and then see how close your guess is. You can use this information to make another guess and continue guessing until you have found the square root (or a close approximation of it). One particularly good way of making guesses is to use Newton's method. Suppose *x* is the number we want to find the square root of, and *g* is the current guess. The guess can be improved by taking $\frac{g + \frac{x}{g}}{2}$ as the next guess.

   Write a program that implements Newton's method. The program is to prompt the user for the value to find the square root of (*x*) and the number of times *n* to improve the guess. Have the user input these on a single line, separated by a comma. Starting with a guess of *x*/2, your program is to loop the specified number of times applying Newton's method and report the final guess, *g*. It is then to subtract that guess from the value of `math.sqrt(x)` and print the absolute value of that difference to show how close the guess is. Please turn in your program in the file `newton.py`.
   [*text*, §3.8, Programming Exercises problem 17, modified]