No starting code for this homework assignment.

At the start of each of these problems, the name of a Python file is given in **blue**: **foo.py**. You should create and save the requested Python program source code in a file with the same name. Also add a comment at the top of each file giving your name.

When finished, upload each **.py** file with the specified name to the Canvas **H3 Assignment** link.  
  
Do each of the following. Create a new PyCharm project and create each of the following files within it, as described below. For the turtle graphics problem, be sure to have **wn.exitonclick()** as the last statement of your script.

**~~[H3-1]~~** ~~(~~**~~rand10.py~~**~~) Write a program that prints out 100 random integers, each in the range 1 through 6.~~

**~~[H3-2]~~** ~~(~~**~~randN.py~~**~~) Write a program that reads an~~ **~~int numRolls~~** ~~from the user, then generates~~ **~~numRolls~~** ~~random integers, each in the range 1 through 6. This simulates rolling a single fair (unbiased) die.~~

~~Use a~~ **~~for~~**~~-loop and the accumulator pattern to add together all these rolls values giving an integer~~ **~~sum~~**~~, then compute the average roll value (~~**~~sum / numRolls~~**~~). Print out this average. What do you expect for this average, as~~ **~~numRolls~~** ~~gets larger and larger?~~

**[H3-3]** (**calcpi2.py**) Here's another way of approximating Pi, using the Madhava-Leibniz formula:

**Pi/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + ...**

Notice how the signs alternate between **+** and **-** for each term of this series (**1, -1/3, 1/5, -1/7,** etc.). Write a program which reads **num\_terms** as an **int**, then computes the above approximation of **Pi/4** by summing the first **num\_terms** terms of this series. Then multiply your computed sum by 4 and print out the resulting approximate calculated value of **Pi** as well the error: the absolute value of its difference with **math.pi**.  
  
Hint: use a loop **for count in range(num\_terms):** initialize **sign=-1.0** followed by a loop body that calculates   
**sign=-sign** and then **term = sign / (2\*count + 1)** for **count=0,1,2...** Accumulate the sum of these terms, using the accumulator pattern with a variable **pi\_estimate**, then calculate and print **4\*pi\_estimate** and its error   
**abs(math.pi-4\*pi\_estimate)**.

**~~[H3-4]~~** ~~(~~**~~http4\_6.py~~**~~) Do Exercise 6 at the end of Chapter 4 in your HTT online book. Hint: you can draw a polygon with~~ **~~n~~** ~~sides by repeating the following~~ **~~n~~** ~~times: move forward the length of a side, then turn left~~ **~~360/n~~**~~. Study the chapter examples to see how to set the color of each side, as well as how to fill your polygon with a given color. You can find all legal colors for setting on the~~ **~~Turtle Colors~~** ~~link next to this handout on Canvas, as well as here:~~ [**~~https://www.tcl.tk/man/tcl8.5/TkCmd/colors.htm~~**](https://www.tcl.tk/man/tcl8.5/TkCmd/colors.htm)

**[H3-5]** (**trace\_me.py**) The Python program **trace\_me.py** is posted next to this handout on Canvas. Download it and add to your project for this assignment. Set breakpoints on the two **print("set...")** statements in this program. Finally, run your program in the PyCharm debugger and enter **5** for the first input, then the values **3.0, 3.1, 3.14, 3.13, 3.141** for the five subsequent inputs.

When your program stops at each breakpoint, write down the values of **a, b,** and **l**. Then resume until the next breakpoint is encountered. Finally, add comments (lines starting with **#**) at the end of the program which lists the five sets of such breakpoint values that you wrote down, followed by the values for the final breakpoint. Your program should thus end with a comment that lists six total sets of the values for **a, b,** and **l.** Submit your code with these comments as your delivery for this problem.