**~~You should submit your own original work for this and subsequent Homework assignments~~**~~. You should use PyCharm to write your Python code. No starting code is provided for this assignment.~~

~~At the start of most of the following, the name of a Python file is given in~~ **~~blue~~**~~:~~ **~~foo.py~~**~~. You should create and save the requested program source code in a file having exactly this name.~~ **~~Please add a comment at the top of each submitted .py giving your name and the name of the source file.~~**

~~When finished, upload each~~ **~~.py~~** ~~file~~ **~~separately~~** ~~to the Canvas~~ **~~H1 Assignment~~** ~~link,~~ **~~as part of a single submission~~**~~. This means all of the following files should be uploaded in the~~ **~~same submission~~**~~.~~

**~~[H1-1]~~** ~~(~~**~~htt2\_1.py~~**~~) Create a folder somewhere on your PC to hold your~~ **~~.py~~** ~~files. Name it~~ **~~H1~~** ~~and use it to hold both this problem's~~ **~~.py~~** ~~file, as well as the subsequent problem files. But... you must submit each .~~**~~py~~** ~~file separately! Easier for me to grade this way...~~

* 1. ~~Open PyCharm, then select Configure->Settings (Configure->Preferences on MacOS) and set up the default Python interpreter as we did in class within Lab 1. A posted Help Video also describes this process. You should only need to do this once for any installed instance of PyCharm.~~
  2. ~~Create a new PyCharm Project using your~~ **~~H1~~** ~~folder by clicking on Open, navigating and selecting this~~ **~~H1~~** ~~folder (NOT any of the~~ **~~.py~~** ~~files therein) and clicking on the Open button.~~
  3. ~~The H1 -...- PyCharm window should appear, with Project view on the left-hand side showing~~ **~~H1~~** ~~and other elements.~~
  4. ~~With~~ **~~H1~~** ~~selected in the Project view, create a new empty Python file~~ **~~htt2\_1.py~~**~~: File->New...->Python File~~
  5. ~~For each expression~~ ***~~expr~~*** ~~in HTT2 Exercise 1, add a~~ **~~print(~~*~~expr~~*~~)~~** ~~to your file. Note that PyCharm automatically saves files while editing.~~
  6. ~~When finished, run your program: Run->Run… and select~~ **~~htt2\_1.py.~~** ~~Your output will appear in a Run window below.~~
  7. ~~Fix any syntax errors until it runs correctly.~~
  8. ~~Observe the output, and make sure you understand how the given expressions evaluate to the given output.~~

**~~[H1-2]~~** ~~(~~**~~eval1.py~~**~~) In HTT2 Exercise 2, the parenthesized expression~~ **~~2 - (3 - 1) \* 10 / 5 \* (2 + 3)~~** ~~is given. This chapter describes the order in which Python evaluates the different sub-expressions within it, leading to a single value for evaluating the entire expression.~~  
~~Write a program which breaks this expression down into a series of assignment statements, each of the form~~ ***~~var = e1 op e2~~***~~, where~~ ***~~var~~*** ~~is some variable, where~~ ***~~op~~*** ~~is one of the arithmetic operators~~ **~~+,-,/,\*~~** ~~and each of~~ ***~~e1~~*** ~~and~~ ***~~e2~~*** ~~are either (a) an~~ **~~int~~** ~~literal or (b) a variable you assigned to in earlier statements. Your final assignment statement should be of the form~~ **~~result~~** ~~=~~ ***~~e1 op e2.~~***

**~~Note that each of your assignment statements should have only ONE operator and TWO operands on the right-hand side.~~**

~~After this, add the two print statements~~ **~~print(result)~~** ~~and~~ **~~print(2-(3-1)\*10/5\*(2+3)).~~** ~~Be sure you evaluate the sub-expressions in the correct order when calculating~~ **~~result~~**~~, so that the output of both is the exactly the same. Note that~~ **~~47~~** ~~is NOT the same as~~ **~~47.0~~**~~!~~

~~Example for the expression~~ **~~44 + 1 - 2 / (3 - 4)~~**~~:~~

**~~temp1 = 3 - 4 # eval most deeply-nested expressions first, from left to right~~**

**~~temp2 = 2 / temp1 # / has higher precedence than + or -, so do it first~~**

**~~temp3 = 44 + 1 # +, - have equal precedence, so eval left to right~~**

**~~result = temp3 - temp2 # final calculation assigned to variable result~~**

**~~print (result) # result should be 47.0~~**

**~~print (44 + 1 - 2 / (3 - 4)) # ...and so should this~~**

**~~[H1-3]~~** ~~(~~**~~eval2.py~~**~~) Same as the previous, but for the expression:~~ **~~1.0 + 2.0 \* 2 \*\* 3 \*\* 2 % 3 - 6 // 4.~~** ~~Note the use of the two operators~~ **~~\*\*~~** ~~(exponentiation) and~~ **~~//~~** ~~(integer division).~~

**~~[H1-4]~~** ~~(~~**~~htt2\_8.py~~**~~) Do HTT2 ("How to Think..." Chapter 2) Exercise 8. Your code should prompt the user to enter the radius of a circle, read the entered value, then calculate and print out its area. Output both the result and a description of what it represents.~~

**~~[H1-5]~~** ~~(~~**~~pytypes.py~~**~~) Python's~~ **~~type(obj)~~** ~~function returns the type of~~ **~~obj~~**~~. Write a program which prints out as many different types as you can, such as:~~ **~~print(type([47])).~~** ~~Note you'll need to do some research on types beyond those mentioned in HTT2.~~