**COMP 1800 – Fall 2016**

**Classwork 15: Using Built-in Math Functions**

**(15 points)**

Number of People: Teams of up to 2. If you work with a teammate, only one submission is needed. Be sure to put both of your names in a comment at the top of each source code file, as well as in the eCourseware notes box when you submit. Feel free to ask me or Swaroop for help!

Due: Nov. 17, by the end of class

Submission: Zip all of your Python script files into a single file, and submit that zip file to the appropriate folder on eCourseware.

Grader: TA, Swaroop Goli ([ssgoli@memphis.edu](mailto:ssgoli@memphis.edu)). Questions about grading? Please contact him first!

**Save your script file as: CW15.py**  
  
In trigonometry there is an identity that states for any value of *x*. In words, this just says

1. Take the sine of any value *x*, and square it
2. Take the cosine of that same *x*, and square it
3. Add the two squared values. You will always get 1; it doesn’t matter what value you picked for *x*.

Write a Python program that verifies this identity by computing and displaying the value of for a wide range of *x* values: *x* = 0.0, 0.1, 0.2, 0.3, and so on, all the way up to *x* = 100.0. Your program should display the value of *x*, followed by the value of . Here’s an example of what the first several lines might look like:  
  
x = 0: result = 1.0

x = 0.1: result = 1.0

x = 0.2: result = 1.0

x = 0.30000000000000004: result = 1.0

x = 0.4: result = 1.0

x = 0.5: result = 1.0

x = 0.6: result = 1.0

x = 0.7: result = 1.0

x = 0.7999999999999999: result = 1.0

x = 0.8999999999999999: result = 1.0

x = 0.9999999999999999: result = 0.9999999999999998

x = 1.0999999999999999: result = 1.0  
  
Note that you may see some very slight errors in the values of *x* and/or the result. This is because floating-point values in Python are not always exact!  
  
Hint: Start by writing a loop that covers the desired range of *x* values. What’s the starting value? What’s the ending value? How much should the value increase each time?