Laboratory Exercise #1

This Lab Exercise assumes:
1. You have read Chapter 1
2. The Pre-Lab Assignment has been completed before attending your lab session.

Assignment Overview

This exercise provides practice with numeric computations in Python.

You will work with a partner on this exercise during your lab session. Two people should work at one computer. Occasionally switch the person who is typing. Talk to each other about what you are doing and why so that both of you understand each step.

Roots of a Quadratic Equation

Consider the quadratic equation:

\[ A \times x^2 + B \times x + C = 0 \]

where \( x \) is the unknown and \( A, B \) and \( C \) are constants (with \( A \) not equal to 0). A quadratic equation has two solutions (called roots), which may not be distinct values and which may not be real values.

The two roots of a quadratic equation may be calculated using the quadratic formula. See the brief article at Wolfram MathWorld if you don’t recall the formula:


Develop a program to compute the two roots of a quadratic equation.

1. Copy the file named “lab01.py” into your account.

2. Modify that program to compute the two roots of a quadratic equation, as described in the program comments. Note that the program does not perform any error checking, so the results displayed by the program may not be correct in all cases. For example, when \( A \) is zero, the equation is not quadratic (e.g. if you get a `ValueError: math domain error`, you have tried to take the square root of a negative value.) We include the `round(number, precision)` function to round the long result to assist with automatic checking of your output by Mimir.

3. Test your completed program using the following values:

   Test1: \( A = 1, B = 0, C = -4 \)
   
   Root #1 = ____________  (should be 2.0)
   
   Root #2 = ____________  (should be -2.0)

   Test2: \( A = 1, B = 5, C = -36 \)
   
   Root #1 = ____________
Root #2 = ____________

Test3: A = 2, B = 7.5, C = 6

Root #1 = ____________
Root #2 = ____________

Test4: A = 0, B = 3.5, C = 8  # this test case fails and generates an error (why?):
   # ZeroDivisionError: float division by zero
   # There is no Mimir test for this case.

Root #1 = ____________
Root #2 = ____________

Test5: A = 5, B = 0, C = 6.5  # this test case fails and generates an error (why?):
   # ValueError: math domain error
   # There is no Mimir test for this case.

Root #1 = ____________
Root #2 = ____________

★ Submit your code to Mimir as lab01.py (the name must be exactly that!) to run the automatic tests. Students in labs: only one submission is needed for students working in pairs; then show your results to your TA – your TA needs to see how Test4 and Test5 run in lab. On-line Section 730 students: your TA will look at your code and the test results on Mimir.

Note (from the syllabus): Labs are credit/no-credit. To receive credit for a lab

1. You must complete the Pre-Lab, before your lab or the pre-lab deadline, whichever comes first. Pre-labs are "warm-up" for the labs and are not expected to be perfect -- our expectation is necessarily fuzzy for pre-labs: "you are expected to get most of them correct most of the time."
2. When Mimir tests exist, they test perfection. The Mimir tests allow you to verify that your code is correct. However, you can get credit for correct Lab code that isn't perfect, i.e. it is possible to get credit for code that fails Mimir tests. (Note that Projects are more strict with respect to Mimir tests.)
3. Adhering to the Coding Standard is expected, but expectations are less strict than for Projects.