**Complex Numbers**

1. Write a class Circle, whose constructor takes a single argument, representing the radius. Put the class definition in a file called circle.py
2. Write a member function, called area, for the class that returns the area of the circle.
3. Write a member function, called diameter, that returns the circle diameter. Write a member function, called circumference that returns the circumference of the circle.
4. Now we're going to write a class for complex numbers. Start by writing a class called Complex that takes two arguments, representing the real and imaginary parts of a complex number. Put the class definition in a file called complex.py, and call the member variables re and im. You should make these arguments optional, and default to something sensible if they are missing. Check that your code does the right thing in response to the following:

a = Complex(1.0, 2.3)    # 1 + 2.3i  
b = Complex(2)           # 1 + 0i  
c = Complex()            # 0 + 0i

1. Implement the \_\_str\_\_() function, so that

a = Complex(1, 2)  
b = Complex(1, -2)  
print a  
print b

works and prints out

(1 + 2i)  
(1 - 2i)

1. Implement addition, such that the following works:

a = Complex(1, 2)  
b = Complex(3, 4)  
print a + b  
print a + 1  
print 1 + a

and prints out

(4 + 6i)  
(2 + 2i)  
(2 + 2i)

Once you have addition working, implement subtraction. This should look a lot like addition.

1. Now, implement multiplication and division. Verify that everything works as expected.
2. Finally, implement the complex conjugate:

a = Complex(1, 2)  
print ~a

should print out

(1 -2i)

1. OK. Let's use your new class. Write a new function that calculates the roots of a quadratic function. You should call it like this:

roots(1, 2, 3)

and it should return a list of all of the roots of the function. Sometimes these will be real numbers, sometimes the will be complex. Sometimes there will be two roots, and sometimes there will be one, or zero, depending on the values you pass in. You should deal with all of these cases.