Fall 2013 CSCI2961: Programming in Python Homework 5

General Instructions

This homework assignment is to be completed individually. Do not share code or review anyone else's code. Work on this assignment is to be your own.

Submit your homework via RPILMS before midnight on the due date. Put all of your code into exactly one Python file and name it with the homework number followed by an underscore, followed by your RCS userid. For example, if your RCS userid is mehtaa2, then your Python file name for this homework assignment must be hw5_mehtaa2.py.

Be sure to comment your code and include your name at the top of each file submitted.

Design a package for managing Length and Area measurements

This assignment is designed to give you experience with the operator overloading and class design capabilities of Python.

There have been a number of serious problems caused by application programs that did not factor in the units of measurement in calculations. For example, NASA lost the $125 million Mars Climate Orbiter in 1999 because one team used English units, while another used metric units (see spacemath.gsfc.nasa.gov/weekly/6Page53.pdf).

There are multiple approaches for incorporating units into calculations, and the general problem of tracking units can be complex. For this assignment, we will explore the design of a simple class library for handling two types of measurements: Length and Area. To further simplify the problem, we will limit the range of units that must be supported to the following:

Length:	m (meter), cm (centimeter), mm (millimeter), km (kilometer), in (inch), ft (foot), yd (yard), mi (mile)
Area	sq_m (square meter), sq_cm, sq_mm, sq_km, sq_in, sq_ft, sq_yd, sq_mi, acres

To avoid calculation errors, each quantity will be maintained as a pair of values (the numeric value and the associated unit of measurement).

You are to design a framework for managing units, quantities, conversion between units, and arithmetic operations involving quantities. This framework is intended for use by other programmers in creating Python programs. Client programmers of your class library will be using these measurements in many complex formulas, so your class library must allow them to write code that is clean, simple, easy to use, and easy to understand.

If a quantity is created using a specific unit, such as “centimeter”, you must track and remember that unit in all calculations and operations. Use operator overloading capabilities to define arithmetic operations for quantities. If you are performing an arithmetic operation using two quantities of **different** units, use the unit on the left side of the expression for the resulting unit (for example, “meters” * “inches” would give you a resulting measured in “meters”).

Define methods that programmers can call to convert a quantity from one unit to another (e.g. “meters” to “inches”).

Your goal is to allow for the following syntax (assuming that the client programmer has specified the necessary imports to define the names “meter”, “centimeter”, …):

```
>>> length1 = 1 * meter; print(length1)
1m

>>> length2 = 2 * meter; print(length2)
2m

>>> length_1_2 = length1 + length2; print(length_1_2)
3m

>>> length3 = 200 * millimeter; print(length3)
200mm

>>> length1_3 = length1 + length3; print(length1_3)
1.2m

>>> print(length1_3.centimeter())
120cm

>>> area = length1 * length2; print(area)
2m^2

>>> area + length1
Traceback (most recent call last):
...
Exception: Unit conversion error in expression: 2m^2 + 1m
```

These are a few examples of the types of operations your code should support, but I have not listed all possible operations in the examples above. Look through the available Python arithmetic operations, to determine which ones are valid and mathematically meaningful to this problem, and make sure that these are supported in your submission.