

Programming Project #10

Assignment Overview

This assignment focuses on the implementation and testing of a Python class to manipulate parallelograms, as well as an application program which uses that class.

It is worth 60 points (6% of course grade) and must be completed no later than 11:59 PM on Monday, April 14.

Assignment Deliverables

The deliverables for this assignment are the following files:

- proj10_quad.py – the source code for your class “Quad”
- proj10_test.py – the source code for your test program
- proj10_app.py – the source code for your application program

Be sure to use the specified file names and to submit them for grading via the **handin system** before the project deadline.

Assignment Background

A parallelogram is a quadrilateral with two pairs of parallel sides. That is, side AB is parallel to side CD, and side BC is parallel to side DA.

The instructors have designed “class Quad” and have supplied an outline of that class in the file named “proj10_quad.py”.

You will complete the development of “class Quad” by implementing each function which currently appears in “proj10_quad.py” as a stub (incomplete function definition). You will demonstrate that your implementation of the class is correct by developing a program which serves as a test bed for that class.

When you have implemented and tested “class Quad”, you will develop an application program which uses that class to read a series of user-supplied parallelograms and then displays information about those parallelograms (see below).

Specifications for Class Quad

1. Your implementation of “class Quad” must be based on the class outlined in “proj10_quad.py” on the course website. That file contains a stub for each method (member function); each stub contains the following line:

```
pass # REPLACE
```

For each method, you will delete that line and replace it with one or more lines of Python source code to complete the method.

2. You may not alter “class Quad” in any other way: you may not alter any line of the file except those lines which contain “pass” statements. This implies that you may not add new methods to the class, nor may you delete existing methods from the class.
3. The constructor will accept three parameters: the length of side AB, the length of side DA, and the degree measure of the angle A between those two sides. If angle A is missing, it will be assumed to be 90 degrees. If side DA is missing, it will be assumed to be the same length as side AB.
4. The constructor will validate the parameters. When side AB, side DA and angle A are all numeric, when side AB and side DA are greater than zero, and when angle A is greater than zero and less than 360, the parallelogram is valid.
5. The representation of a valid object of type “Quad” will be a string containing the length of each side. That string will be in the form “[x.x, x.x, x.x, x.x]”, where the square brackets and spaces are required. Each “x.x” represents the length of one side; each length will be displayed with exactly one digit to the right of the decimal point. For example: [7.2, 133.8, 7.2, 133.8].
6. The method “sides” will return a tuple containing the four sides.
7. The method “angles” will return a tuple containing the degree measure of each of the four interior angles (with exactly one digit to the right of the decimal point), if the parallelogram is valid. Otherwise, it will return a tuple containing four instances of None.
8. The method “perimeter” will return the perimeter, if the parallelogram is valid. Otherwise, it will return zero.
9. The method “area” will return the area, if the parallelogram is valid. Otherwise, it will return zero.
10. The method “scale” will multiply each of the sides by the given factor, if the parallelogram is valid and the factor is greater than zero. Otherwise, it will not alter the four sides.
11. None of the methods in “class Quad” will use function “print”. That is, they will not display messages to the user, even if an error is detected.

Specifications for the Test Program

1. You will develop a program to serve as a test bed for “class Quad”. That is, the only purpose of the program is to demonstrate that each method of “class Quad” is implemented correctly.

The source code for your test bed will be contained in the file named “proj10_test.py”. That file will import “proj10_quad.py”.

2. Your test bed will not perform any input operations. Instead, all test cases will be embedded in the program itself.
3. The output produced by your test bed will be appropriately labeled so that the reader can understand the purpose and result of each test case without examining the source code.

Specifications for the Application Program

1. You will develop an application program which uses “class Quad” to solve the problem described below.

The source code for your application program will be contained in the file named “proj10_app.py”. That file will import “proj10_quad.py”.

2. The program will attempt to access the file named “quad_input.txt”. If that input file cannot be opened, the program will display an appropriate message and halt.

3. The input file will contain zero or more lines, where each line contains at most one parallelogram.

A parallelogram will appear as a string in the form “ x, y, x ”, where “x” represents the length of side AB, “y” represents the length of side DA, and “z” represents the degree measure of angle A. The double quotes will not be present and the whitespace is optional. For example: 1.25, 45,60.

4. The program will read each line of the input file and display it. The display will be on one line, will begin with “Line N:” (where N is the current line number) and will contain the exact contents of the line. Line numbers will begin with 1.

The program will then display two more lines: a line with information about the input line and a blank line. If an input line contains a valid parallelogram, the program will display the parallelogram as a string, and then display its perimeter and area. If an input line does not contain a valid parallelogram, the program will display an appropriate message.

5. After the program has processed all of the lines in the input file, it will display the following:

- a. The total number of lines processed.
- b. The total number of valid parallelograms processed.
- c. The average perimeter for all valid parallelograms processed.
- d. The average area for all valid parallelograms processed.
- e. The valid parallelogram with the largest perimeter (the parallelogram, its perimeter and area).
- f. The valid parallelogram with the largest area (the parallelogram, its perimeter and area).

If more than one parallelogram has the largest perimeter or largest area, display the first such parallelogram which is found in the input file.

Assignment Notes

1. As noted under “Specifications for Class Quad”, the only lines in “proj10_quad.py” which you may alter are the lines containing “pass # REPLACE”. Each of those lines should be deleted, and you should substitute one or more lines to complete the functionality of the class.

2. Please note that you may wish to use function “print” to display various items as you are developing your implementation of the class. However, all invocations of function “print” must be removed from the final version of your class (or at least turned into comments).

3. You would be wise to develop your implementation of “class Quad” and your test program incrementally and in parallel. That is, implement one class method at a time and then test that method by adding statements to your test program.

Clearly, the first class method which must be implemented and tested is the constructor (“__init__”). However, you might choose to first implement the constructor without worrying about erroneous parameters (error handling could be added later).

Perhaps the second class method to implement and test is “__str__” so that you have a way to display the value of an object of type “Quad” using function “print”.

Then, you might be wise to implement and test a relatively simple method, such as “perimeter”.

After implementing and testing those three methods, you would continue to implement and test one method at a time until you have completed the class.

Be sure to insert and test any necessary error handling at the appropriate time. For example, it makes no sense to add error handling to method “perimeter” until the constructor detects invalid parameters.

4. Approximately 50% of the 60 points available for the project will be allocated to the implementation of “class Quad”, approximately 25% will be allocated to the test program, and approximately 25% will be allocated to the application program.