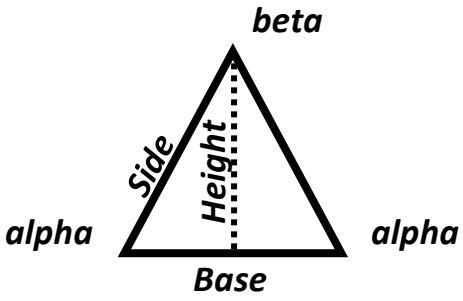
EE-3233

Systems Programming for Engineers

Lab Assignment 4

Your task is to design a Python3 class that models an isosceles triangle. However, this is by no means a simple class. The novelty of this implementation lies on its ability to automatically calculate and update multiple parameters on the fly, which are derived from the triangle's base and height.



This assignment has a high difficulty level, start early.

**** This document is 3 pages long. ****

API

Your class should implement an API as follows:

Data Attribute	Description
Base	Only modifiable by __init__, set_base (int or float).
Height	Only modifiable by __init__, set_height (int or float).
Side	The length of each lateral side (float).
Perimeter	The triangle's perimeter (float).
Area	The triangle's area (float).
Alpha	Angle between each side and the base (degrees in float).
Beta	Angle between the two sides (degrees in float).

Method	Description
__init__	Takes two parameters, width and height. To build a new rectangle. This method automatically updates all the triangle's data attributes.
set_base	Change the current length value. New length must be greater than 0. This method automatically updates all the triangle's data attributes.
set_height	Change the current height value. New height must be greater than 0. This method automatically updates all the triangle's data attributes.
calc_side	Compute and return the current value of each of the two lateral sides of the triangle (You only return one value).
calc_perimeter	Compute and return the current value of the perimeter.
calc_area	Compute and return the current value of the triangle's area.
calc_alpha	Compute and return the current value of the angle alpha.
calc_beta	Compute and return the current value of the angle beta.

Tips

The hardest part of implementing the class is calculating the length of each lateral side. To do so, use the Pythagorean theorem by using the functions provided by the math module. Here, you will have to import the **`math`** module as follows:

```
import math
```

And within your code, you can get the square root of a number, **`math.sqrt()`**, as well as other mathematical constants and trigonometric functions. You will definitely have to make use of the **`math.atan()`** function, and the **`math.degrees()`** function to convert from radians. Additionally, elevating a number to the power of n is done as follows:

```
y = x ** 2 # y equals to x to the power of 2.
```

Finally, remember that you can implement more methods in addition to the ones described by the API. These may help you simplify your code.

Testing

Define the following method in the triangle class (you can just copy-paste this):

```
def print_all(self) -> None:
      print(f"------------------------------")
      print(f"base     : {self.base}")
      print(f"height   : {self.height}")
      print(f"side     : {self.side}")
      print(f"perimeter: {self.perimeter}")
      print(f"area     : {self.area}")
      print(f"alpha    : {self.alpha}")
      print(f"beta     : {self.beta}")
      print(f"------------------------------")
```

This method will be used by another script to test your implementation.

Deliverables

You must upload the script with your solution, named with the following format:

triangle_<first-name>_<last-name>.py

For example:

triangle_andres_hernandez.py

Report-like submissions in PDF format will no longer be accepted.

Grading

- Your solution will be evaluated with a script to check if your solution is correct.
- Partial marks will be granted for cases where the output is partially correct.
- Late submission will be subject to 10% off per day

Furthermore, following the syllabus, your script will be evaluated using Code Similarity Checker utilities to detect similar code and/or plagiarism.