

Project 3 – The New Triangle Program

You are to write a variant of the triangle program. Instead of taking three integers as input, this program will take in the length of sides A and B, as well as the angle C in degrees. (Assume A, B, and C must be given in integers, and $0 < C \leq 178$). Your program will then use the Law of Cosines to determine the length of C, and then output one of the following statements:

- ⇒ Your triangle is equilateral (if all three sides are equal)
- ⇒ Your triangle is isosceles (if two but not three sides are equal)
- ⇒ Your triangle is right (if one angle is right)
- ⇒ Your triangle is obtuse (if one of the angles is greater than 90°)
- ⇒ Your triangle is scalene (if none of the above conditions are true)

(Given the length of sides A and B and the angle of C, you can learn about how to use the Law of Cosines to determine the length of C as well as the angles of A and B at this website:

<https://www.mathsisfun.com/algebra/trig-cosine-law.html>.)

Your program must be structured with three methods (or procedures, subroutines, or functions, depending on the language you are using). Each time the user inputs the three variables (length of A, length of B, and the angle C):

1. The first method (called FindLengthOfC) should be called once
2. The second method (called FindAngle) will be called either once or twice, depending on how you structure the method and its calling code
3. The third method (called DetermineTriangleType) will be called once

Note: If you prefer, you can use underscores for the method names instead of CamelCase (e.g., *Find_Length_of_C* or *determine_triangle_type* would also be acceptable method names in this software – just be consistent with your naming conventions).

The program should also print out the dimensions of the triangle. For example, if the user input was 7, 7, 60, the program should print:

- ⇒ The dimensions of your triangle are 7,7,7 (60, 60, 60); your triangle is equilateral.

The non-integer calculated lengths and angles should be rounded to the nearest tenth.

As we did with Project 2, the project will be completed in milestones, and each milestone will be a graded homework assignment.

HW 3a: Test Plan. Write a test plan for your triangle program.

HW 3b: Code. Write the code for your program. If you want to use TDD for this step, you can do so, but that is not required for this project.

HW 3c: Flow Graph. Create a flow graph for your third method. Do a flow graph analysis to determine how many test cases would need to be run in order to achieve statement and branch coverage (path coverage is optional).

HW3d (a.k.a. Proj 3): White Box Test Plan. Compare your original test plan with your flow graph analysis, and determine if any test cases need to be added to your test plan in order to attain branch coverage. If so, add these test cases. Then run the complete test plan and document the results.