# Computer Science

UBC

SFU
SIMON FRASER UNIVERSITY

Student:

Tommy Ju

Teacher:

Ms. Lawrence

# Table of Contents

# Formal Proposal

- Formal Proposal Word Document

# Capstone Timeline

Due Date: December 13 2019

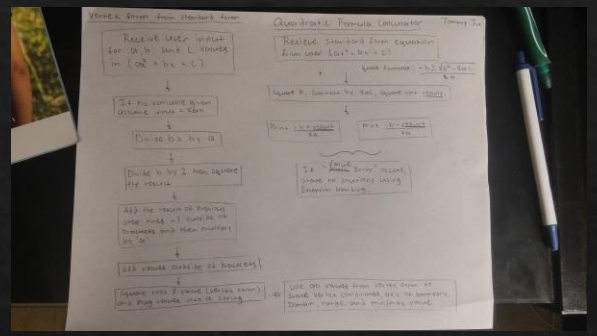
Oct. 17 – Research and develop flow charts for Vertex form converter and Quadratic Formula Calculator.

Oct. 30 – Code and debug program using flow charts as references.

Nov. 2 – Research modules that can be used to solve triangles and create a flowchart for the Triangle Solver program.

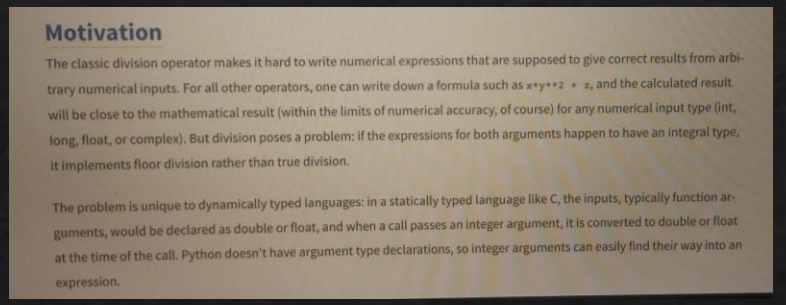
Dec. 11 – Finish Coding and debugging the Triangle Solver, create a GUI if done planning before Nov. 20

# Logs for Vertex and Quadratic Calculators



Created flow charts for Vertex form converter and Quadratic Calculator. Also reviewed procedures for completing the square.

**10 Oct.**




Had trouble with Python's "Classic division", researched and learned how to use "division" function from __future__ module to get accurate numbers.

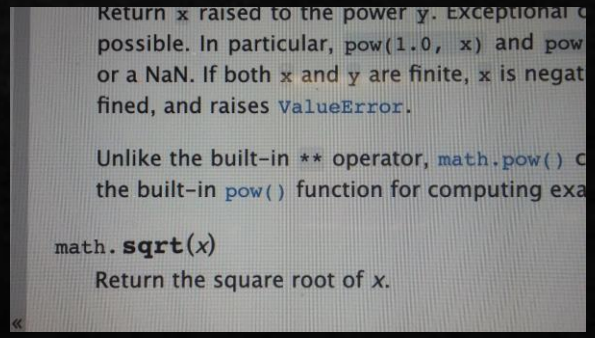
20 Oct.

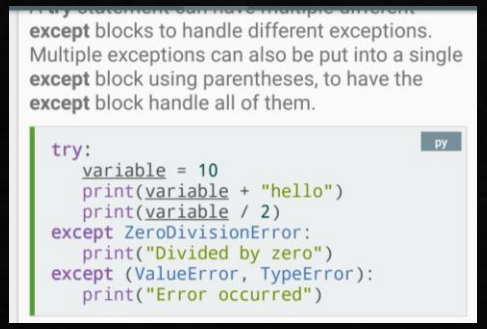
Finished debugging and making sure the Vertex Form Converter works properly. Added extra features to find slope given coordinates, x – intercepts, and y – intercept.

**27 Oct.**

16 Oct.

Learned about the square root function under the Math module and learned how to use Exception Handling for the Quadratic Calculator.

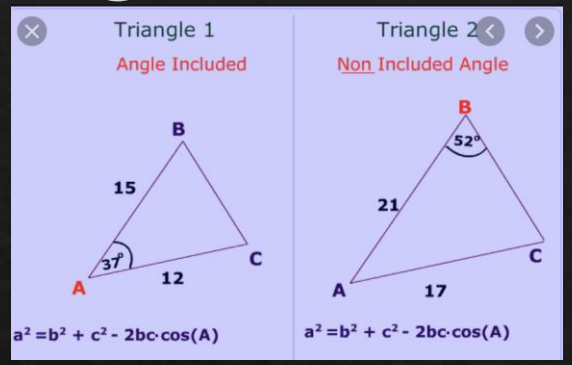



25 Oct.

Finished debugging and coding the Quadratic Formula Calculator

# Logs for Triangle Solver Program (2/3)

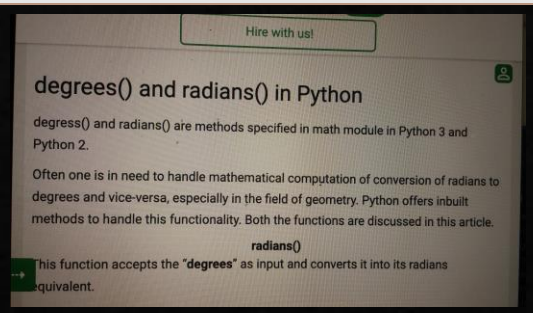



Added more detail to program's flowchart and learned about non-included and included angles when solving SSA triangles.

23 Nov.

Learned about degrees() and radians() functions which can be used to convert degrees to radians and vice versa.

1 Dec.




Code didn't work properly, figured out a way to turn a function's input into a list rather than a tuple.

2 Dec.

Learned how to use the round() function in Python.

3 Dec.

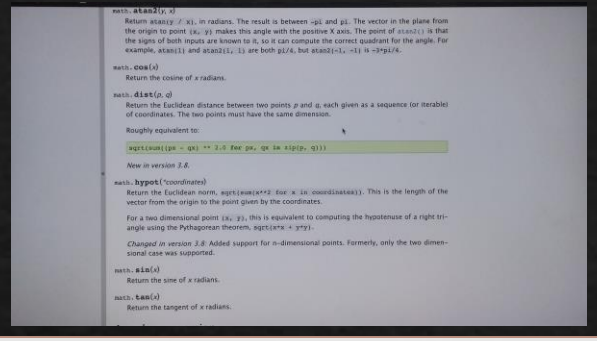
**Python round() Function**

The **round() function** returns a floating point number that is a rounded version of the specified number, with the specified number of decimals. The default number of decimals is 0, meaning that the **function** will return the nearest integer.

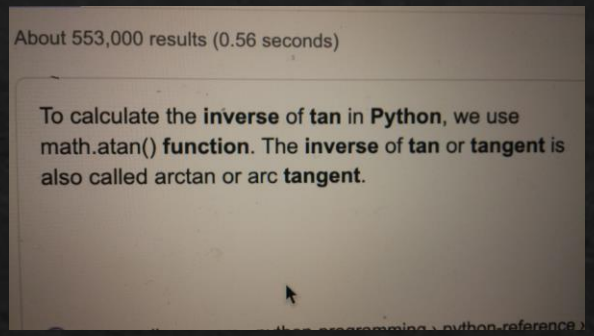
Ran into a very tedious bug, Python's sin() and cos() functions only work in radians and my code was using degrees.

5 Dec.

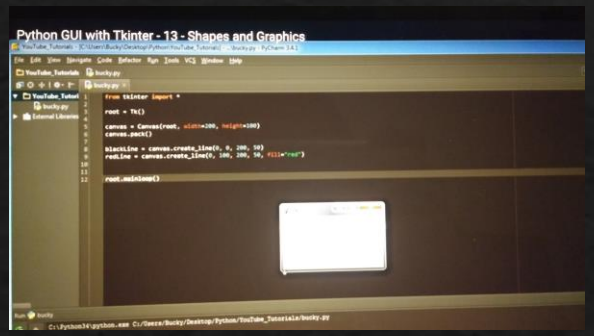
# Logs for Triangle Solver Program (1/3)




Researched and found trigonometric functions under the Math module that can return an answer in radians on the Python official website.



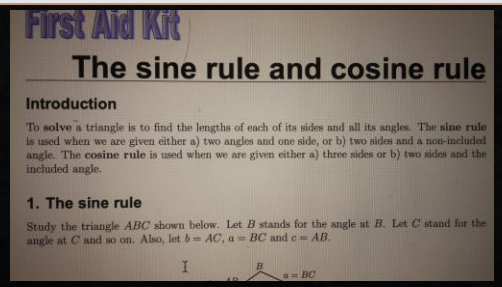

Learned about arc tangent, arc sine, and arc cosine functions that can be used to find the angle of a trigonometric ratio.



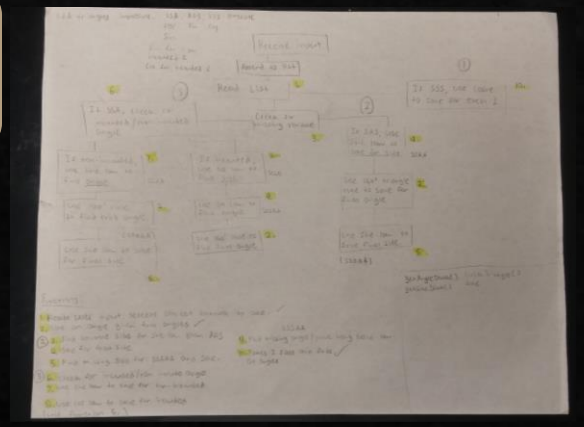

Taught myself how to make a GUI using the Tkinter module from videos on Youtube.

6 Oct. — 2 Nov. — 5 Nov. — 11 Nov. — 18-23 Nov.

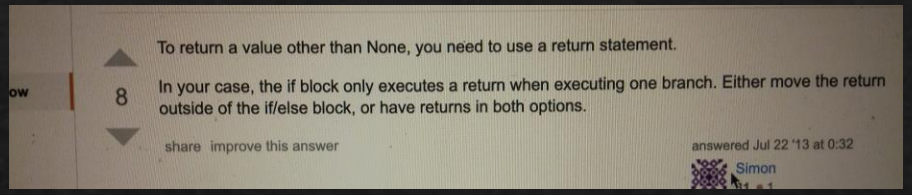
Reviewed how to use cosine and sine laws to solve triangles (SSA, SAA, SSS), and developed a rough flow chart for program.




Expanded on the program design by mapping out functions that the program can use to solve various triangles.



# Logs for Triangle Solver Program (3/3)

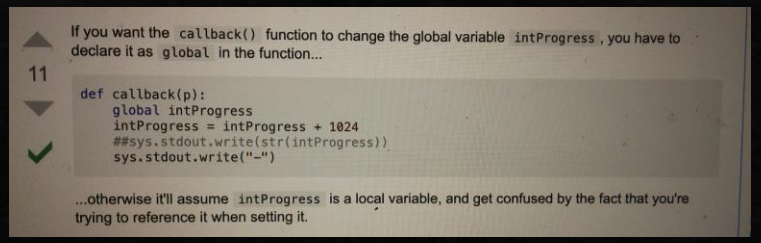



Spent over 15 hours coding and debugging over the weekend.

6-7 Dec.

11 Dec.

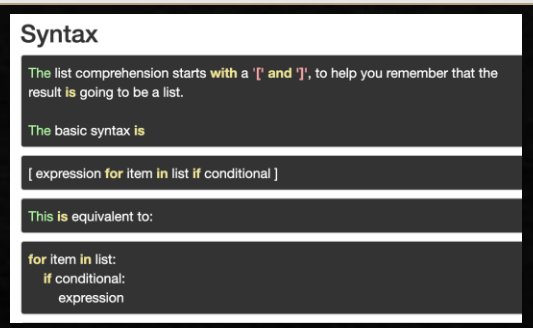
Learned how to reference a global variable inside a function and solved a bug by learning that you always have to return a nested function.




When packing a widget in Tkinter, you have to pack it separately on a new line or else creates a "NoneType".

11 Dec.

Learned how to use "list comprehension" to iterate through a list and overwrite its elements.




Finished creating a program that accounts for every possibility to solve any triangle with 3 data points and at least one side.

12 Dec.

# Goals

- Goals Word Document

# Resume and Cover letter

- Dropbox link

# Reference Letters

- Dropbox link

# Final Reflection

- Capstone Final Reflection

# Core Competencies

- Personal Stories Assignment
- Self Assessment of Core Compentencies