

```

# This program calculates the input values from the user to
# determine whether it forms a right triangle.
# Try-except are used to catch errors in the process.
# Created by Virginia Gonzalez
# November 30th, 2022

import sys
import math

class RightTriangle:
    # This function initializes all the sides of a right triangle
    def __init__(self, side_a, side_b, side_c='0'):
        try:
            self.side_a = float(side_a)
            self.side_b = float(side_b)
            if side_c == '0':
                self.side_c = math.sqrt(self.side_a ** 2 + self.side_b ** 2)
            else:
                self.side_c = float(side_c)
        except:
            # first exception is raised when the values cannot be casted to a float
            raise ValueError('Error: Invalid numbers.')
        if self.side_a <= 0 or self.side_b <= 0 or self.side_c <= 0:
            # second exception is raised if the values are negative
            raise ValueError('Error: Both sides and the hypotenuse must be non negative.')
        if math.fabs(self.side_a ** 2 + self.side_b ** 2 - self.side_c ** 2) > 0.01:
            # third exception is raised if the values entered do not form a right triangle
            raise ValueError('Error: Sides a, b, and c do not form a valid Pythagorean triplet.')

    def __eq__(self, other):
        result = math.fabs(self.side_c - other.side_c)
        # returns true if the two sides of the other triangle are very close to the two sides of this triangle,
        # including if the other triangle's side a is very close to this triangle's side b and vice-versa
        if result <= 0.01:
            return True
        # Returns false if the hypotenuse of the other RightTriangle is not very close to equal to
        # the hypotenuse of this RightTriangle
        else:
            return False

    def __str__(self):
        return f'Right Triangle with side a = {self.side_a:.1f}, side b = {self.side_b:.1f}, '\
               f'and hypotenuse = {self.side_c:.1f}'

def main():

```

```
print()
print('This program calculates the input values from the user to ')
print('determine whether it forms a right triangle.')
print()
righttriangle = 0
righttriangle2 = 0
while True:
    try:
        a = input('Enter side a: ')
        b = input('Enter side b: ')
        righttriangle = RightTriangle(a, b)
        print(righttriangle)
        break
    except ValueError as value:
        print(value)
while True:
    try:
        a = input('Enter side a: ')
        b = input('Enter side b: ')
        c = input('Enter side c: ')
        righttriangle2 = RightTriangle(a, b, c)
        print(righttriangle2)
        break
    except ValueError as value:
        print(value)
print()
print('First triangle: ')
print(righttriangle)
print()
print('Second triangle: ')
print(righttriangle2)
print()
print('Does the first triangle equal to itself?')
print(righttriangle == righttriangle)
print('Does the first triangle equal to the second?')
print(righttriangle == righttriangle2)
```

```
# Press the green button in the gutter to run the script.
```

```
if __name__ == '__main__':
    sys.exit(main())
```

This program calculates the input values from the user to determine whether it forms a right triangle.

```
Enter side a: -1
Enter side b: 2
Error: Both sides and the hypotenuse must be non negative.
Enter side a: 2
Enter side b: e
Error: Invalid numbers.
Enter side a: 2
Enter side b: 3
Right Triangle with side a = 2.0, side b = 3.0, and hypotenuse = 3.6
Enter side a: -3
Enter side b: 4
Enter side c: r
Error: Invalid numbers.
Enter side a: 2
Enter side b: 3
Enter side c: 4
Error: Sides a, b, and c do not form a valid Pythagorean triplet.
Enter side a: 3
Enter side b: 4
Enter side c: 5
Right Triangle with side a = 3.0, side b = 4.0, and hypotenuse = 5.0

First triangle:
Right Triangle with side a = 2.0, side b = 3.0, and hypotenuse = 3.6

Second triangle:
Right Triangle with side a = 3.0, side b = 4.0, and hypotenuse = 5.0

Does the first triangle equal to itself?
True
Does the first triangle equal to the second?
False
```