16 Aug 2019

ANL 251 Python Programming

A toy application
(Supplementary readings)

Kevin Kam Fung Yuen, PhD

Senior Lecturer

School of Business

Singapore University of Social Sciences

kfyuen@suss.edu.sg,
kevinkf.yuen@gmail.com

Use Cases for the The story

1. Create a Heron module
2. Record the history how users use this applications
3. Change the user name in the db.

Heron module

- Create a Heron module (**Heron.py**) with a function calculate the valid inputs of lengths of 3 sides, a, b, c.
- Create a **testHeron.py** to test the function in the module

https://en.wikipedia.org/wiki/Heron%27s_formula

Heron.py

```python
import math

# create a Heron function
def Heron_triagle_area(sides):
    """
    calculate triangle area
    attributes: a iterable set [a, b, c] or (a, b, c)
                of non zero and positive value
    return: triangle area
    """
    # Implement the logic to check the validity of code in the future
    a = sides[0]
    b = sides[1]
    c = sides[2]
    s = (a + b + c) / 2
    A_square = s * (s - a) * (s - b) * (s - c)
    A = None
    if A_square > 0:
        A = math.sqrt(A_square)
    else:
        print(f"The inputs {sides} cannot form a triangle.")
    return A
```

Heron's formula states that the area of a triangle whose sides have lengths a, b, and c is

$$A = \sqrt{s(s-a)(s-b)(s-c)},$$

where s is the semi-perimeter of the triangle; that is,

$$s = \frac{a+b+c}{2}.$$[2]

Let $\triangle ABC$ be the triangle with sides $a = 4$, $b = 13$ and $c = 15$. The semiperimeter is $s = \frac{1}{2}(a+b+c) = \frac{1}{2}(4+13+15) = 16$, and the area is

$$A = \sqrt{s(s-a)(s-b)(s-c)} = \sqrt{16 \cdot (16-4) \cdot (16-13) \cdot (16-15)}$$
$$= \sqrt{16 \cdot 12 \cdot 3 \cdot 1} = \sqrt{576} = 24.$$

In this example, the side lengths and area are all integers, making it a Heronian triangle. However, Heron's formula works equally well in cases where one or all of these numbers is not an integer.

```python
# testHeron.py
# test the function in Heron module
from Heron import Heron_triagle_area
test = Heron_triagle_area([4, 13, 15])
print("The expected result should be 24: ", test)

test1 = Heron_triagle_area((0, 1, 15))
print("error?: ", test1)
```

```
PS D:\_0SUSS\ANL251Python\MyCode\L4> python testHeron.py
The expected result should be 24:  24.0
The inputs (0, 1, 15) cannot form a triangle.
error?:  None
```

Record the history how users use this application

queryAndRecord.py

```python
from Heron import Heron_triagle_area
import datetime

print("Welcome")
user_name = input("What is your name? ")
a = float(input("input a positive number for side 1 : "))
b = float(input("input a positive number for side 2 : "))
c = float(input("input a positive number for side 3 : "))
submit_time = str(datetime.datetime.now())

area = Heron_triagle_area([a, b, c])

print("The area is ", area)
print("Bye!")

# keep record
record = user_name +" " + str(submit_time) + " " \
         + str([a, b, c]) + " " + str(area) + "\n"

file = open("log.txt","a")
file.write(record)

file.close()
```

log.txt

```
Kevin 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
Kevin 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
Kevin 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
```

```
PS D:\_0SUSS\ANL251Python\MyCode\L4> python queryAndRecord.py
Welcome
What is your name? Kevin
input a positive number for side 1 : 6
input a positive number for side 2 : 8
input a positive number for side 3 : 10
The area is  24.0
Bye!
```

log.txt has been updated.

New Record is added

```
Kevin 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
Kevin 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
Kevin 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
Kevin 2019-08-16 16:49:03.078752 [6.0, 8.0, 10.0] 24.0
```

New Record

Change the user name in the log.

changeUserName.py

```python
# assume the name exist
print("Change old_name")
old_name = input("What is the name to be changed? ")
new_name = input("What is the new name? ")

file1 = open("log.txt","r")
data = file1.readlines()

temp = []
count = 0
for line in data:
    if line.find(old_name) != -1:
        print("we find one")
        count += 1
        temp.append(line.replace(old_name, new_name))
    else:
        temp.append(line)

file2 = open("log.txt","w")
file2.writelines(temp)

file1.close()
file2.close()

print(f"we change {count} {old_name} to {new_name}")
print("bye")
```

```
Kevin 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
Kevin 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
Kevin 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
Kevin 2019-08-16 16:49:03.078752 [6.0, 8.0, 10.0] 24.0
```

```
PS D:\_0SUSS\ANL251Python\MyCode\L4> python changeUserName.py
Change old_name
What is the name to be changed? Kevin
What is the new name? Kevin YUEN
we find one
we find one
we find one
we find one
we change 4 Kevin to Kevin YUEN
bye
PS D:\_0SUSS\ANL251Python\MyCode\L4>
```

```
Kevin YUEN 2019-08-16 15:42:40.053516 [2.0, 4.0, 6.0] None
Peter 2019-08-16 15:43:18.932160 [4.0, 6.0, 7.0] 11.976539567003485
Alice 2019-08-16 15:43:30.147776 [3.0, 3.0, 5.0] 4.14578098794425
Kevin YUEN 2019-08-16 15:43:53.687989 [7.0, 8.0, 9.0] 26.832815729997478
Kevin YUEN 2019-08-16 15:44:17.342385 [3.0, 35.0, 10.0] None
John 2019-08-16 15:44:36.580056 [4.0, 7.0, 9.0] 13.416407864998739
Alice 2019-08-16 15:44:53.384703 [50.0, 70.0, 90.0] 1741.228014936585
Kevin YUEN 2019-08-16 16:49:03.078752 [6.0, 8.0, 10.0] 24.0
```