# Homework 3

## your name and id

This homework has two problems. Please fill the code block cells with your code and comments, ***run*** everything (select `cell` in the menu, and click `Run all` ), save the notebook, and upload it to canvas.

## Problem 1: Triangle Class ¶

Define the class named `Triangle` , which takes three positive number $a, b, c$ as the input to initialize the instance, representing the length of three sides. The class should have

- three attributes named a,b,c corresponding to the input.
- a method named `is_triangle` , which returns `True` if the three number forms a valid triangle, and `False` othersie.
- a method named `perimeter` , which returns the perimeter of triangle.
- a method named `area` , which returns the area of triangle. *Hint: you can use* [Heron's formula](https://www.mathsisfun.com/geometry/herons-formula.html) *(https://www.mathsisfun.com/geometry/herons-formula.html)*
- a method named `height` , which returns a list containg the three heights corresponded to the three input bases.

In [1]:

```python
import math

class Triangle:
    '''write your codes below (don't forget to change this doc strings)'''

    def __init__(self,a,b,c):
        '''write your codes below (don't forget to change this doc strings)'''

    def is_triangle(self):
        '''write your codes below (don't forget to change this doc strings)'''

    def perimeter(self):
        '''write your codes below (don't forget to change this doc strings)'''

    def area(self):
        '''write your codes below (don't forget to change this doc strings)'''

    def height(self):
        '''write your codes below (don't forget to change this doc strings)'''
```

When your code is done, run it and test the class with following piplines -- Please re-run and save the results below.

In [3]:

```python
triangle_1 = Triangle(1.0,2.0,3.0)
triangle_1.is_triangle()
```

Out[3]:

```
False
```

In [4]:

```python
triangle_2 = Triangle(3.0,4.0,5.0)
print(triangle_2.is_triangle())
print(triangle_2.perimeter())
print(triangle_2.area())
print(triangle_2.height())
```

```
True
12.0
6.0
[4.0, 3.0, 2.4]
```

Note that your defined methods should be valid for any triangle instead of special ones.

## Problem 2: Line Class

Define the class called `Line` which represents a line with equation $y = kx + b$ with input slope $k$ and intercept $b$ to initialize the instances. It should include:

- attributes named $k$ and $b$ to represent slope and intercept.
- method named `intersect` to return the list, containing coordinates of the intersection point of two lines.
- support for `+` operator to compute the addition of two equations. The sum of two `Line` objects $y = k_1x + b_1$ and $y = k_2x + b_2$ is defined as the line $y = (k_1 + k_2)x + b_1 + b_2$.
- printable representation for the equation of line, which we have already defined in `__repr__` speical method below.

In [5]:

```
class Line:
    '''write your codes below (don't forget to change the doc strings)'''

    def __init__(self, k, b):
        '''write your codes below (don't forget to change the doc strings)'''

    def __repr__(self):
        '''printable representation of the line by equation y = k*x+b'''
        return 'line y = %r*x + %r' % (self.k, self.b)

    def __add__(self,other):
        '''write your codes below (don't forget to change the doc strings)'''

    def intersect(self,other):
        '''write your codes below (don't forget to change the doc strings)'''
```

When your code is done, run it and test the class with following piplines -- Please re-run and save the results below.

In [7]:

```
line1 = Line(2,1)
line2 = Line(1,2)
line3 = line1+line2
```

In [8]:

```
line3
```

Out[8]:

```
line y = 3*x + 3
```

In [9]:

```
point = line1.intersect(line2)
print(point)
```

```
[1.0, 3.0]
```