HONG KONG INSTITUTE OF VOCATIONAL EDUCATION
Laboratory 3a: Instance and Static Members

TASK:

1. Estimate the output of the following code segment without using computer. Verify your result by using computer after you have finish it.

```
class Test:
    test = 10
    def reset_test(self):
        self.test = 0
    def add_one(self):
        self.test += 1

if __name__ == "__main__":
    c1 = Test()
    c1.add_one()
    print (f"c1.test = {c1.test}")
    print (f"Test.test = {Test.test}")
    c1.reset_test()
    c1.add_one()
    print (f"c1.test = {c1.test}")
    print (f"Test.test = {Test.test}")
```

Estimated output:
c1.test = 11
Test.test = 10
c1.test = 1
Test.test = 10

2. (a) Create a class **Rectangle** that has:

Rectangle
__length : float __width : float __no_of_rectangle : int
__init__ (self, length, width) find_area (self) : float find_perimeter (self) : float find_number_of_rectangle () : int

Attributes:

length – to store the length of rectangle
width – to store the width of rectangle
no_of_rectangle – to store the total number of rectangle created by rectangle class

Methods:

__init__ – initialize the length and wdith attributes
adds no_of_rectangle by one to count the total numbers of rectangles created
find_area – return the area of rectangle object in float format
find_perimeter – return the perimeter of rectangle object in float format
find_number_of_rectangle – return the no_of_rectangle static variable in integer format

Answer:

```
class Rectangle:
    __number_of_rectangle = 0
    def __init__(self, length, width):
        self.__length = length
        self.__width = width
        Rectangle.__number_of_rectangle += 1
    def find_area(self):
        return self.__length * self.__width
    def find_perimeter(self):
        return (self.__length + self.__width) * 2
    @staticmethod
    def find_number_of_rectangle():
        return Rectangle.__number_of_rectangle
```

(b) Write the supplier class and client / driver program that will do the following:
(i) Create a rectangle object r1 with length 1 and width 1.
(ii) Create a rectangle object r2 with length 2 and width 3.
(iii) Create a rectangle object r3 with length 1 and width 3.
(iv) Print the area and perimeter of the three rectangle objects as shown in the sample output.
(v) Print the total number of rectangles created counted by the static variable __no_of_rectangle.

Sample Output:
r1: area=1, perimeter=4
r2: area=6, perimeter=10
r3: area=3, perimeter=8
Totally there are 3 rectangle(s) created
Answer:

```
if __name__ == "__main__":
    r1 = Rectangle(1, 1)
    r2 = Rectangle(2, 3)
    r3 = Rectangle(1, 3)
    print (f"r1: area={r1.find_area()}, perimeter={r1.find_perimeter()}")
    print (f"r2: area={r2.find_area()}, perimeter={r2.find_perimeter()}")
    print (f"r3: area={r3.find_area()}, perimeter={r3.find_perimeter()}")
        print (f"Totally there are {Rectangle.find_number_of_rectangle()} rectangle(s) created")
```