1	2	3	4	5	Σ

Informatics 2, 2nd midterm (2019-03-08)

The answers should fit next to the questions, if you used a separate paper let us know clearly!

1. Write two python classes which represent points and triangles on the plane. The required methods: *(6 points)*

- `Point`

  constructor with two parameters: the x and y coordinates.

  str method for printing, return the string reprezentation of a point in a format like this: `P(3,4)`

- `Triangle`

  constructor with three paremeters: the three vertex of the triangle as `Point` objects. Raise a `ValueError` if all the points coincide.

  centriod return the centriod of the triangle as a `Point` object. The centroid is the avarage of the vertex vectors.

```
class Point(object):
    def __init__(self,         ):




    def __str__(self           ):


class Triangle(object):
    def __init__(self,         ):




    def centriod(self           ):
```

If you do it correctly the result should work like this:

```
>>> A, B, C = Point(0,0), Point(1,0), Point(0,1)
>>> print B
P(1,0)
>>> h = Triangle(A, B, C)
>>> print h.centriod()
P(0.33333, 0.33333)
```

2. Theoretical questions *(4 points)*

a) How to write your own exception class?

b) What is the `__mul__` special method for? How to use it?

c) How to make a string right justified (for example in the width of 10 characters)?

d) What is a *variadic function*?

3. Write an iterable class that can iterate up to a given number with a given step size! *(3 points)*

The constructor should have two parameters, the number which is the end of the iteration and the step size. The iteration should start at 0 and don't include the end number. Write the `__iter__` and `next` methods accordingly. The `__iter__` method should return `self`!

```
class Iterable(object):
    def __init__(self, n, k):


    def __iter__(self):



    def next(self):
```

Example

```
for i in Iterable(10,2):
    print i,

0 2 4 6 8
```

4. What does the following functions do? *(3 points)*

Explain and give an example!

```
def f(s, width=10):
    return s.center(width)

def g(L):
    R = ""
    for l in L:
        R += f(l)
    return R

def h():
    x = raw_input()
    L = []
    while len(x) > 0:
        L.append(x)
        x = raw_input()
    print g(L)
```

5. Find the 4 mistakes in the following code! *(4 points)*

```
class A(object):
    def__init__(self, x):
        self.x = x

class B(object):
    def __init__(self, *x)
        self.x = x

    def add(self, other):
        return B(self.x + other.x)

    def __str__(self):
        return str(x)
```

Example:

```
a = A(0)
b = B(0)
print b + b
```