

Opgave 1.

- a) Gegeven zijn de onderstaande classes.

```
class A:
    def __init__(self, a_int):
        self.g = a_int
        self.b = None

    def add(self):
        self.g += 1

class B:
    def __init__(self, an_a_obj):
        self.g = 3 * an_a_obj.g
        self.a = an_a_obj

    def subtract(self):
        self.g -= 1
```

en een module waarin de classes A en B en de volgende variabelen en functie staan

```
a = A(1)
b1 = B(a)

def show(an_a_obj, a_b_obj):
    print '%d - %d - %d' % (an_a_obj.g, a_b_obj.g, a_b_obj.a.g)
```

Wat is nu de uitvoer van de hierna gegeven opdrachten die in dezelfde module staan als de gegeven classes, variabelen en functie?

```
show(a, b1)
a.add()
show(b1.a, b1)
b2 = B(b1.a)
show(a, b1)
a.b = b2
show(b2.a, b2)
b1.a.add()
b2.a.add()
show(a, a.b)
b2.subtract()
show(b2.a, b1)
a.b = b1
show(b2.a, b1)
a.b.subtract()
show(a, b1)
```

- b) Geef m.b.v. de functie range() opdrachten waarmee de onderstaande lists gegenereerd kunnen worden.

```
1- [1, 2, 3, 4, 5]
2- [10, 13, 16, 19]
3- [12, 8, 4, 0]
```

Welke strings worden door de onderstaande slices gegenereerd als s gelijk is aan '54647484'?

```
4- s[1:8:2]
5- s[0::2]
6- s[-1:0:-2]
```

- c) Schrijf voor  $n \geq 0$  de functie: `bereken(x, n)`, waarbij `x` van het type `float` en `n` van het type `int` is. De functie rekent de sommatie uit van de eerste `n` termen van de onderstaande berekening

$$(5x^2)/1! - (6x^3)/2! + (7x^4)/3! - (8x^5)/4! + (9x^6)/5! - \dots$$

Waarbij  $x^n$  de notatie is voor "`x` tot de macht `n`" en `n!` voor "de faculteit van `n`" (`n!` is gedefinieerd als:  $1 \times 2 \times 3 \times 4 \times \dots \times (n-1) \times n$ ). Gebruik geen functies uit de module `math`.

- d) Gegeven is het onderstaande programma.

```
e = -5
f = 6

def show (x, y):
    print '%d %d' % (x, y)

def f1 (e, d):
    e *= 2
    f = d + e
    show(e, f)
    return d+1

def f2 (e):
    global f
    show(e, f)
    e = f1(2, e)
    show(e, f)
    f += 3
    return e-f

show(e, f)
f = f1(f, e)
show(e, f)
e = f2(f)
show(e, f)
```

Geef de uitvoer van dit programma.

## Opgave 2.

a) gegeven zijn de onderstaande classes.

```
class Vak:
    def __init__(self, a_str, a_float):
        self.naam = a_str
        self.cijfer = a_float

class Student:
    def __init__(self, a_str, a_int, a_vak_list):
        self.naam = a_str
        self.jaar = a_int # hoeveelste jaar ingeschreven als student
        self.vakken = a_vak_list # een list van Vak-objecten

class Universiteit:
    def __init__(self):
        self.student_list = [] # een list van Student-objecten
```

Schrijf voor de class Universiteit een methode voeg\_toe() waarmee 1 Student-object kan worden toegevoegd aan een Universiteit-object.

b) Voorzie de class Universiteit van een methode

```
def topstudenten (self)
```

welke in een nieuw Universiteit-object de topstudenten retourneert. Voor deze opgave is gedefinieerd dat een student een topstudent is, als het gemiddelde van de cijfers van de vakken van deze student 9,0 of hoger is.

Indien je een deelprobleem herkent, programmeer dit dan in een aparte methode in de juiste class. Gebruik constanten indien nodig.

c) Voorzie de class Universiteit van een methode

```
def gaat_goed (self)
```

welke het aantal studenten retourneert die het goed doen in hun studie. Voor deze opgave is gedefinieerd dat een student het goed doet als 95% (of meer dan 95%) van de vakken als cijfer een voldoende ( $\geq 5,5$ ) heeft.

Indien je een deelprobleem herkent, programmeer dit dan in een aparte methode in de juiste class. Gebruik constanten indien nodig.

d) Gegeven is dat de class Universiteit een methode

```
def ingeschreven (self, jaar)
```

bevat. Deze methode retourneert, in een nieuw Universiteit-object, alle studenten die voor het 'jaar'-de jaar zijn ingeschreven aan de universiteit. Omdat de methode ingeschreven() gegeven is, hoeft deze niet geprogrammeerd te worden om te kunnen gebruiken.

Programmeer nu, zonder gebruik te maken van een for- of while-statement, in de class Universiteit, een methode

```
def gaat_goed_en_ingeschreven (self, jaar)
```

die retourneert met hoeveel van de studenten, die voor het 'jaar'-de jaar zijn ingeschreven aan de universiteit, het goed gaat met de studie.

e) Schrijf voor de class Universiteit een methode

```
def verwijder (self, a_str1, a_str2)
```

welke bij de student met naam 'a\_str1' het vak met de naam 'a\_str2' verwijderd.

Indien je een deelprobleem herkent, programmeer dit dan in een aparte methode in de juiste class. Gebruik constanten indien nodig.

Programmeer efficiënt. Gebruik bij eventuele for- of while-statements niet meer iteraties dan nodig.

Gegeven is dat het eerste voorkomen van een element e in een list x, kan worden verwijderd met de opdracht `x.remove(e)`.

Opgave 3.

Programmeer een functie

```
def verschuif_matrix (m):
```

die voor een willekeurige matrix m van int's, de int's in iedere rij van die matrix als volgt verschuift. Als de rijen van de matrix k kolommen bevatten, worden de int's, in de laatste k-1 kolommen, 1 plaats naar links verschoven en wordt de int in de eerste kolom naar de laatste kolom verschoven.

voorbeeld:	na verschuiven	1	2	3	4	is het	2	3	4	1
	van alle rijen	5	6	7	8	resultaat	6	7	8	5
	in de matrix	9	10	11	12		10	11	12	9
		13	14	15	16		14	15	16	13

Waardering:

opgave	a	b	c	d	e	totaal
1.	5	4	4	5		18
2.	1	4	4	2	3	14
3.	4					4
						-- +
						36

Het eindcijfer E volgt uit het puntentotaal T als volgt:  $E = T / 4 + 1$