

Introduction to Programming

A first year programming course using Python



Peter van Nieuwenhuizen

6 July 2018

The course

- Introductory programming course using Python
- 1st year mathematics students
- 6 ECTS credit points (168 hours)

Applied Mathematics curriculum

	1 ^e Kwartaal	2 ^e Kwartaal	3 ^e Kwartaal	4 ^e Kwartaal		
1 ^e Jaar	Wiskundige Structuren		Modelleren A	Modelleren B		
	Caleidoscoop	Inleiding Programmeren	Algebra 1			
	Lineaire Algebra 1		Analyse 2			
	Lineaire Algebra 1	Analyse 1	Eerstejaars Keuzevak	Inleiding Kansrekening		
2 ^e Jaar	Lineaire Algebra 2		Modelleren 2A	Modelleren 2B		
	Optimalisering		Numerieke Methoden 1			
	Reële Analyse		Partiële Differentiaalvergelijkingen			
	Inleiding Statistiek	Gewone Differentiaalvergelijkingen	Tweedejaars Keuzevak	Complexe Functietheorie		
3 ^e Jaar	Minor		Bachelorcolloquium	Bachelorafstudeerproject		
			Derdejaars Keuzevak			
			Derdejaars Keuzevak			
Modelleren	Numerieke Methoden en Differentiaalvergelijkingen	Discrete Wiskunde	Analyse	Stochastiek	Optimalisering	Keuzevakken

Learning objectives

Students are able to

- write **well structured programs** in Python
 - split a problem into sub problems
 - implement functions to solve the sub problems
- use the **NumPy and Matplotlib** libraries
- write **object oriented programs** in Python

Contents

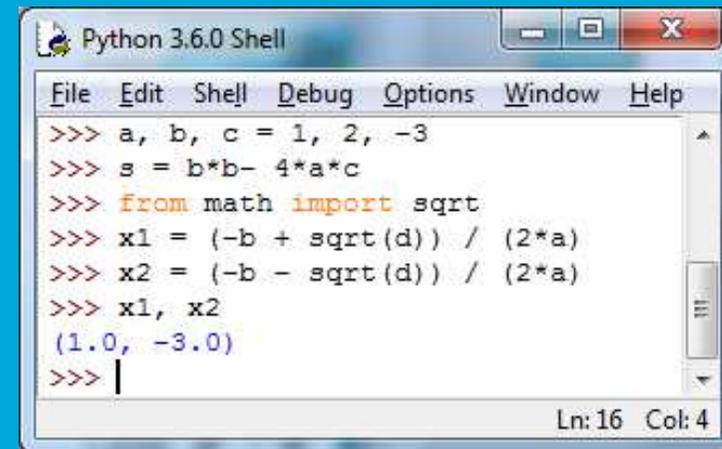
- Programming language Python
- Lectures
- Lab assignments
- Examination

Python

- General purpose programming language
- Low threshold
- 2 methods of working:
 - Interactive in Python shell window
 - Type program code in file; run program

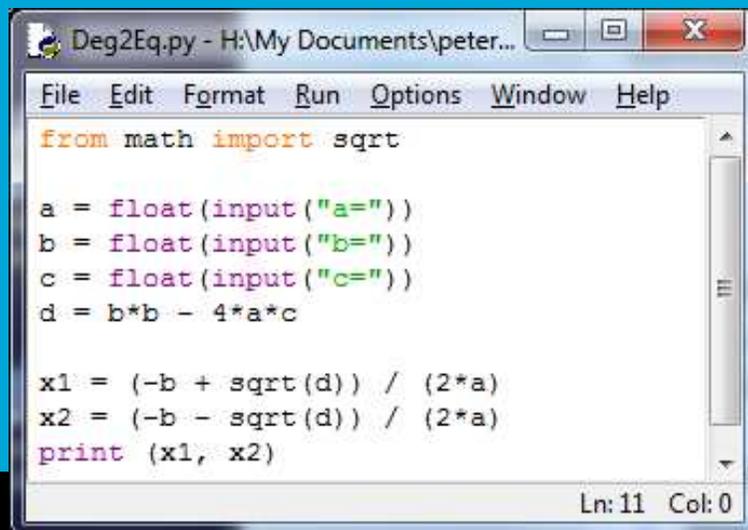
Modes in Python

- Interactive in Python shell window



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
>>> a, b, c = 1, 2, -3
>>> s = b*b - 4*a*c
>>> from math import sqrt
>>> x1 = (-b + sqrt(d)) / (2*a)
>>> x2 = (-b - sqrt(d)) / (2*a)
>>> x1, x2
(1.0, -3.0)
>>> |
Ln: 16 Col: 4
```

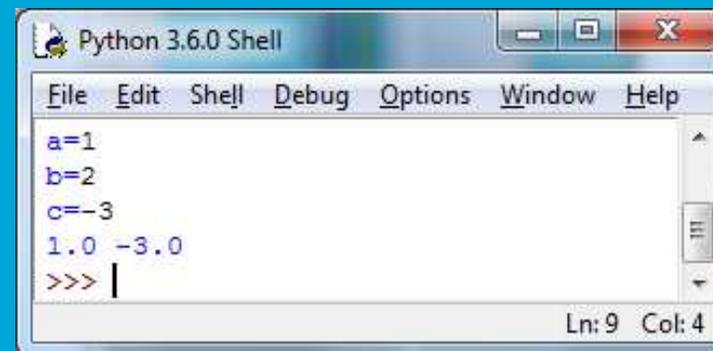
- Type program code in file; run program



```
Deg2Eq.py - H:\My Documents\peter...
File Edit Format Run Options Window Help
from math import sqrt

a = float(input("a="))
b = float(input("b="))
c = float(input("c="))
d = b*b - 4*a*c

x1 = (-b + sqrt(d)) / (2*a)
x2 = (-b - sqrt(d)) / (2*a)
print(x1, x2)
Ln: 11 Col: 0
```



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
a=1
b=2
c=-3
1.0 -3.0
>>> |
Ln: 9 Col: 4
```

Python example 1

```
*Example1.py - H:\My Documents\peter\Windows\TW1090\TW1090slides\Slides2018_Python3\InterTUStu...
File Edit Format Run Options Window Help
points = [(1, 2), (2, 5), (1, 2, 3)]

for point in points:
    print (point)
    print ("sum of coordinates is ", sum(point))
Ln: 2 Col: 0
```

Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
(1, 2)
sum of coordinates is 3
(2, 5)
sum of coordinates is 7
(1, 2, 3)
sum of coordinates is 6
>>> |
Ln: 11 Col: 4
```

Python example 2: Greatest Common Divisor

```
*Example2.py - H:\My Documents\peter\...
File Edit Format Run Options Window Help
def GCD(a, b):
    while b != 0:
        a, b = b, a % b
    return a

print(GCD( 42, 30))
print(GCD(-42, 30))
print(GCD( 42, -30))
print(GCD(-42, -30))
Ln: 9 Col: 8
```

Output

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
6
6
-6
-6
>>> |
Ln: 9 Col: 4
```

Python example 3: Solving a system of linear equations

Output

```
*basic_Numpy_exercises4a.py - H:\My Documents\peter\Windows\TW1090\slides\T...
File Edit Format Run Options Windows Help
import numpy as np

A = np.mat("0, 2, -1; 3, 0, 2; -1, -1, 1")

b = np.mat([[1], [9], [0]])

print "solve Ax=b"
x = np.linalg.solve(A, b)

print "A="
print A
print "b="
print b
print "x="
print x

Ln: 6 Col: 0
```

```
Python Shell
File Edit Shell Debug Options Windows Help

solve Ax=b
A=
[[ 0  2 -1]
 [ 3  0  2]
 [-1 -1  1]]
b=
[[1]
 [9]
 [0]]
x=
[[ 1.]
 [ 2.]
 [ 3.]]
>>>

Ln: 18 Col: 4
```

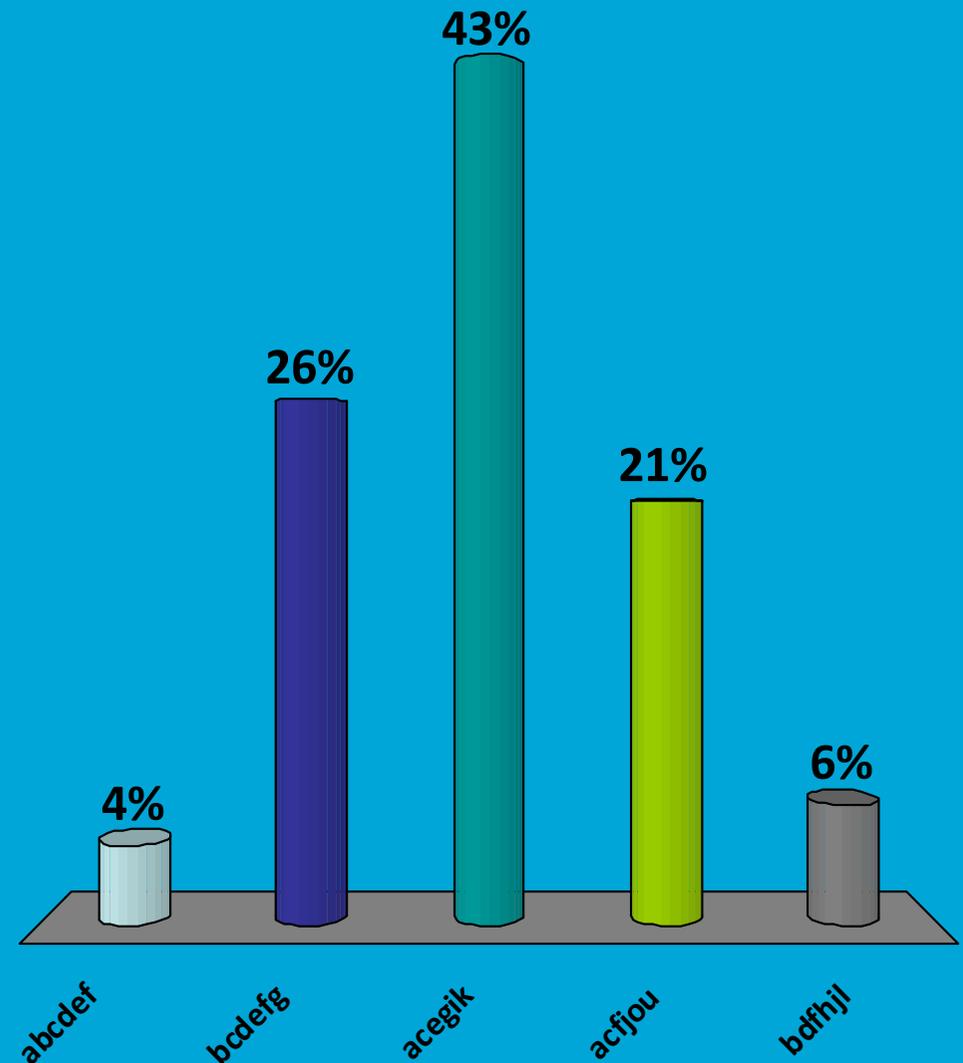
Lectures

- Partly interactive
 - Multiple choice questions (Turning Point)
 - Assignments

What will be printed?

```
TP3.py - H:/My Documents/peter/Windows/TW1090/slides/T...
File Edit Format Run Options Window Help
st = ''
string = 'abcdef'
for i, j in enumerate(string):
    st = st + chr(ord(j) + i)
print st
Ln: 2 Col: 16
```

- A. abcdef
- B. bcdefg
- C. acegik
- D. acfjou
- E. bdfhjl



What will be printed?

```
*TP3.py - H:/My Documents/peter/Windows/TW1090/slides/...
File Edit Format Run Options Window Help
st = ''
string = 'abcdef'
for i, j in enumerate(string):
    print 'i = ', i, 'j = ', j
    st = st + chr(ord(j) + i)
    print 'st = ', st
print st
Ln: 6 Col: 21
```

```
Python 2.7.10 Shell
File Edit Shell Debug Options Window Help
>>>
i = 0 j = a
st = a
i = 1 j = b
st = ac
i = 2 j = c
st = ace
i = 3 j = d
st = aceg
i = 4 j = e
st = acegi
i = 5 j = f
st = acegik
acegik
>>>
Ln: 24 Col: 4
```

Participant Leaders

Points	Participant	Points	Participant
5	36A65D	4	13BA56
5	36A691	4	36A659
5	36A6A1	3	36A65B
4	36A6CA	3	36A661
4	36A6CB	3	36A669
4	36A6E2	3	36A681
4	36A6E7	3	36A68E
4	36A751	3	36A69F
4	36A784	3	36A6D3
4	36A785	3	36A6D4

Assignment

- Start IDLE
- Perform the following calculations in interactive mode (i.e. in a Python shell window)

$$z = 1 - i \quad w = \sqrt{3} + i \quad res = \frac{z^6}{w^3}$$

What is the value of res?

Check your result with a calculation on paper

Advanced:

```
import cmath  
help("cmath")
```

Find out how to get the polar form of z and w

Assignment (solution)

$$z = 1 - i$$

$$w = \sqrt{3} + i$$

$$res = \frac{z^6}{w^3}$$

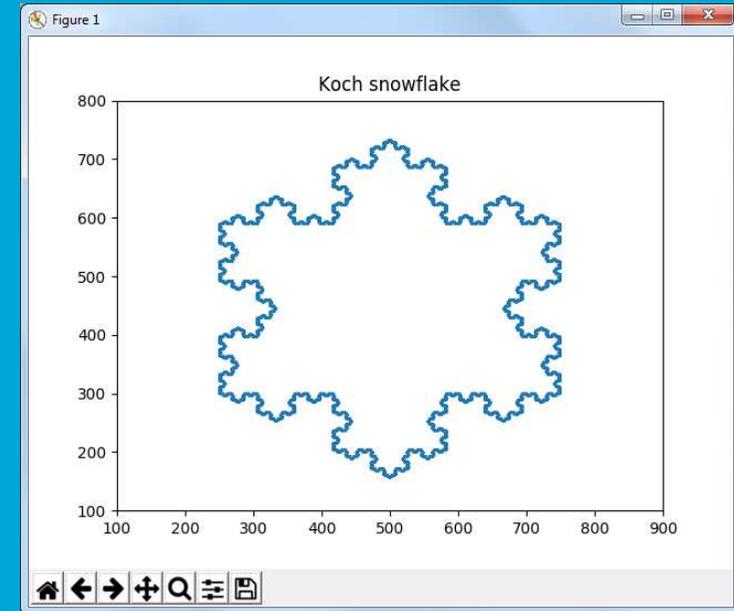
```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
>>> import math
>>> z = 1 - 1j
>>> w = math.sqrt(3) + 1j
>>> res = z**6 / w**3
>>> res
(1.0000000000000002-1.110223024625157e-16j)
>>>
>>> import cmath
>>> print(cmath.polar(z))
(1.4142135623730951, -0.7853981633974483)
>>> print(cmath.polar(w))
(1.9999999999999998, 0.5235987755982989)
>>> |
```

$$z = \sqrt{2}e^{-\frac{1}{4}\pi i}$$

$$w = 2e^{\frac{1}{6}\pi i}$$

Lab assignments

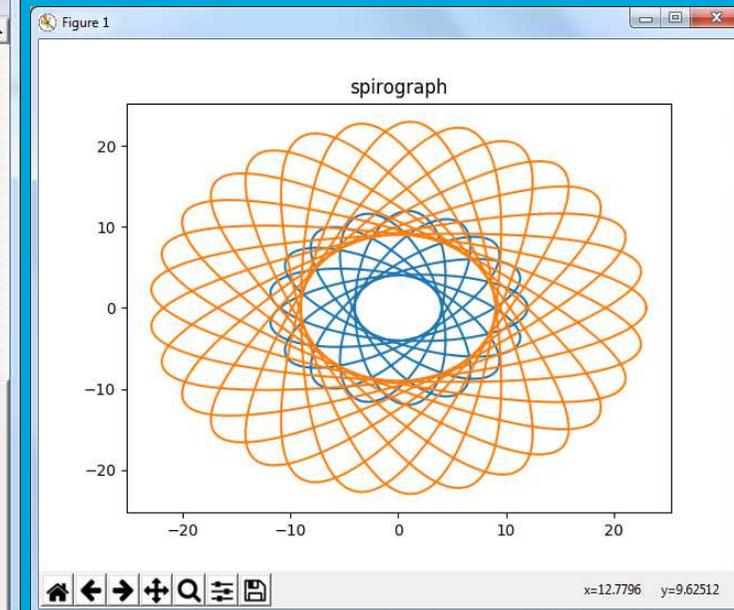
- Examples:
 - Koch snowflake
 - Spirograph
 - Master mind



```
Python Shell
File Edit Shell Debug Options Windows Help
Play mastermind
There are 4 pins and 6 different colors
Pins with different colors are numbered 1,2,...,6
After 10 tries you have lost the game.

guess 4 different colors: 1234
1 on correct place, 1 with correct color.
guess 4 different colors: 1356
1 on correct place, 2 with correct color.
guess 4 different colors: 1465
0 on correct place, 2 with correct color.
guess 4 different colors: 6253
0 on correct place, 4 with correct color.
guess 4 different colors: 2536
4 on correct place, 0 with correct color.
The correct row is: 2536
You have won!

Do you want to play once more? [y or n]:
```

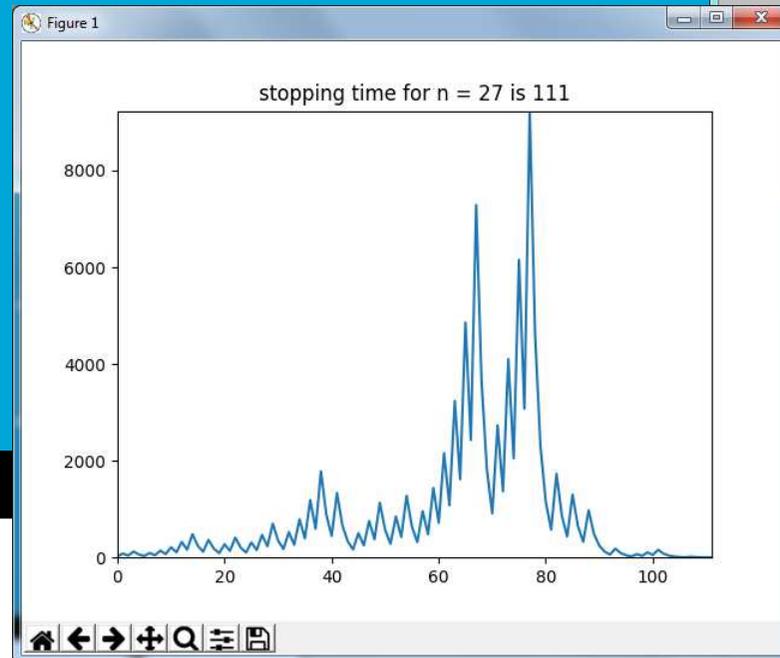
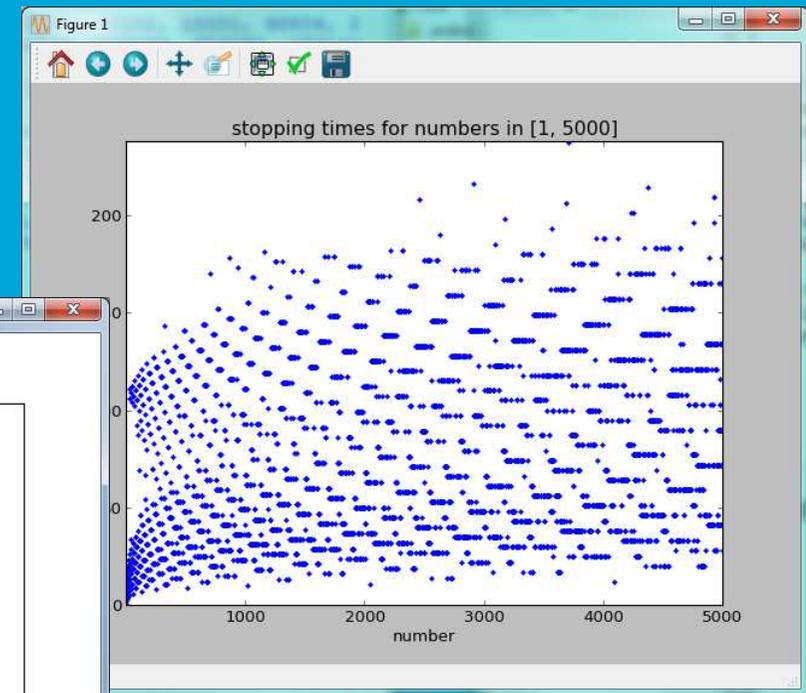
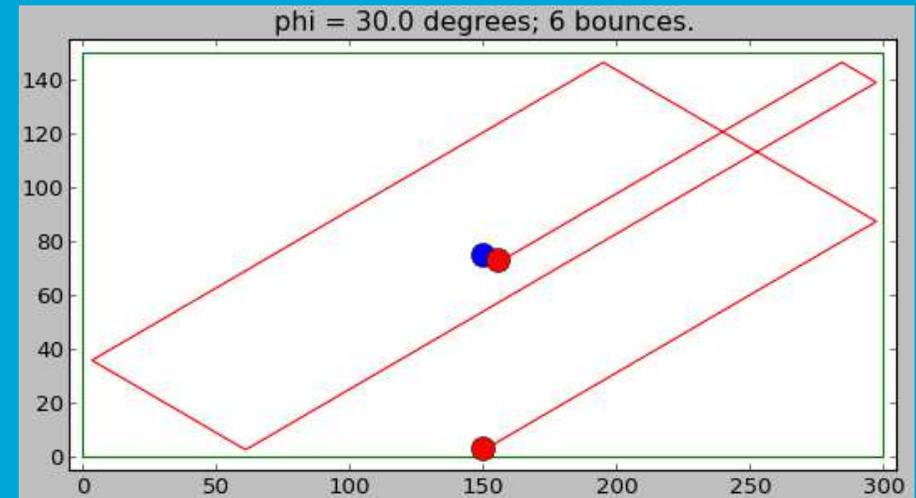


Lab assignments

- Examples:
 - Billiards simulation
 - Collatz conjecture
 - Puzzle

left(A), right(D), up(W),
down(S): d

1	2	3	4	5
6	7	8	9	10
11	12		19	14
16	17	13	18	15
21	22	23	24	20



Advanced parts of lab assignments

Part 1 and 2

Implement a class **Puzzle**

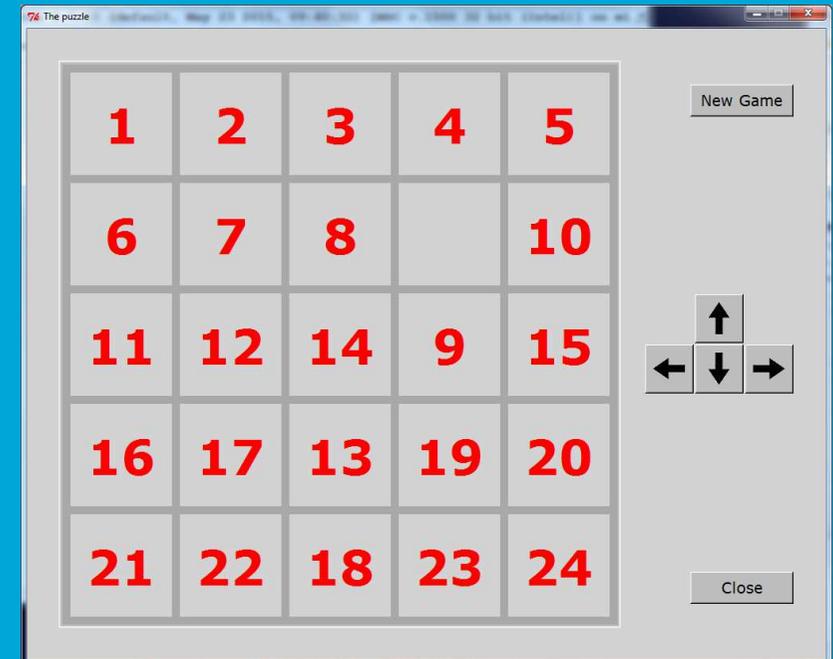
```
left(A), right(D), up(W),
```

```
down(S): w
```

```
1  2  3  4  5
6  7  8  9 10
11 12 14 9 15
16 17 13 18 15
21 22 23 24 20
```

Advanced part 3

Integrate the class **Puzzle** in a program that uses Tkinter



Lab assignment: Collatz conjecture

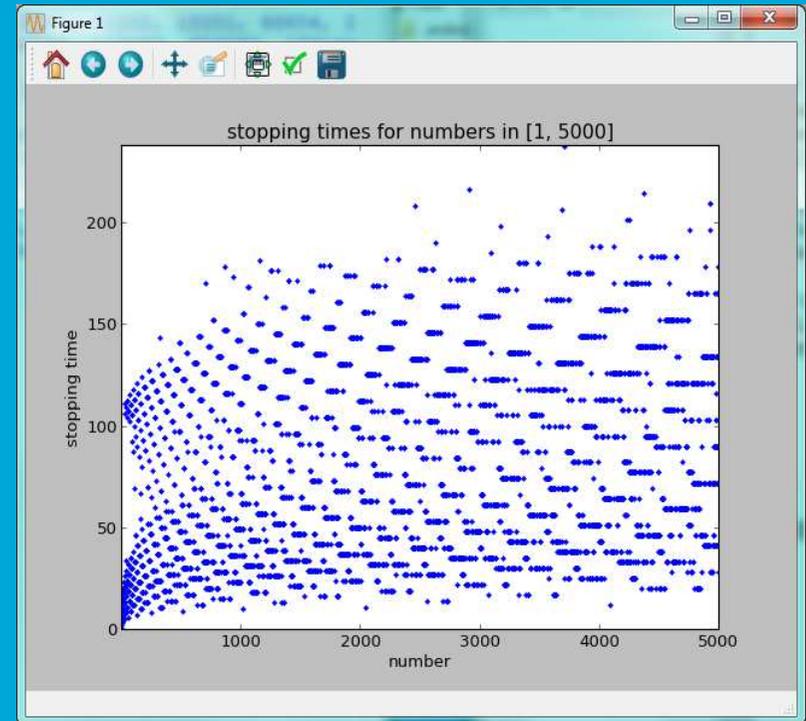
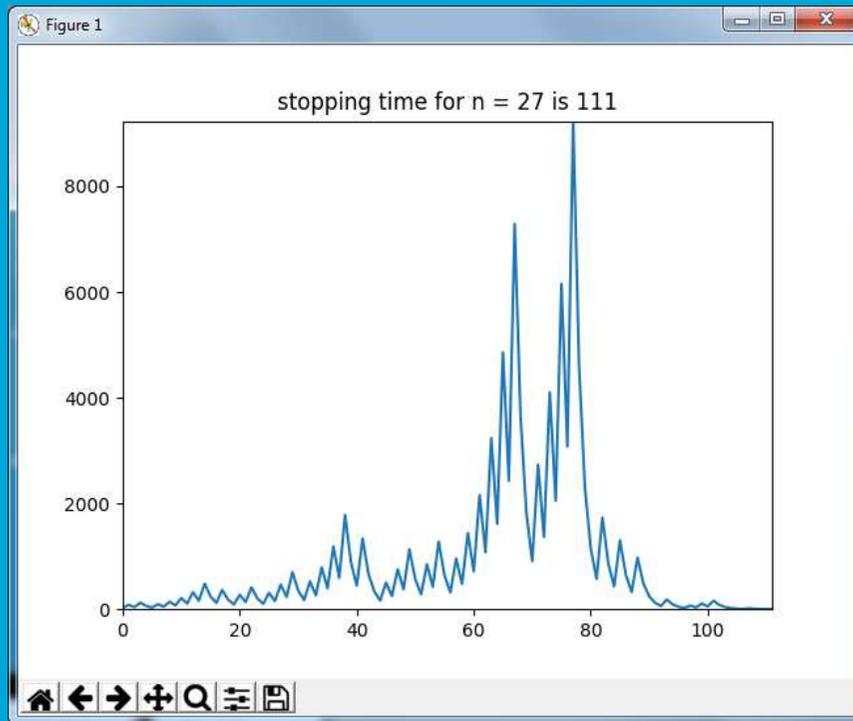
$$a_0 = n$$
$$a_i = a_{i-1} / 2 \quad \text{if } a_{i-1} \text{ even}$$
$$a_i = 3a_{i-1} + 1 \quad \text{if } a_{i-1} \text{ odd}$$

Example:

$n = 3$

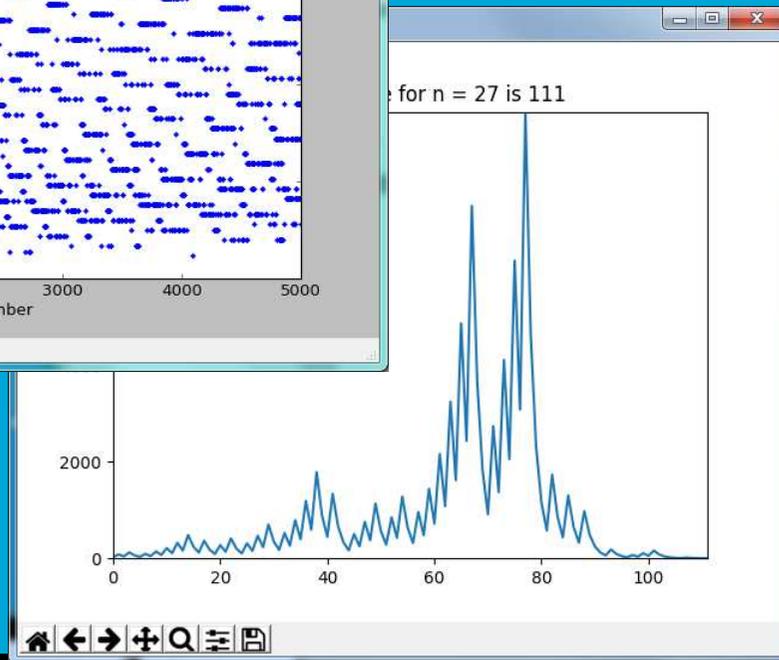
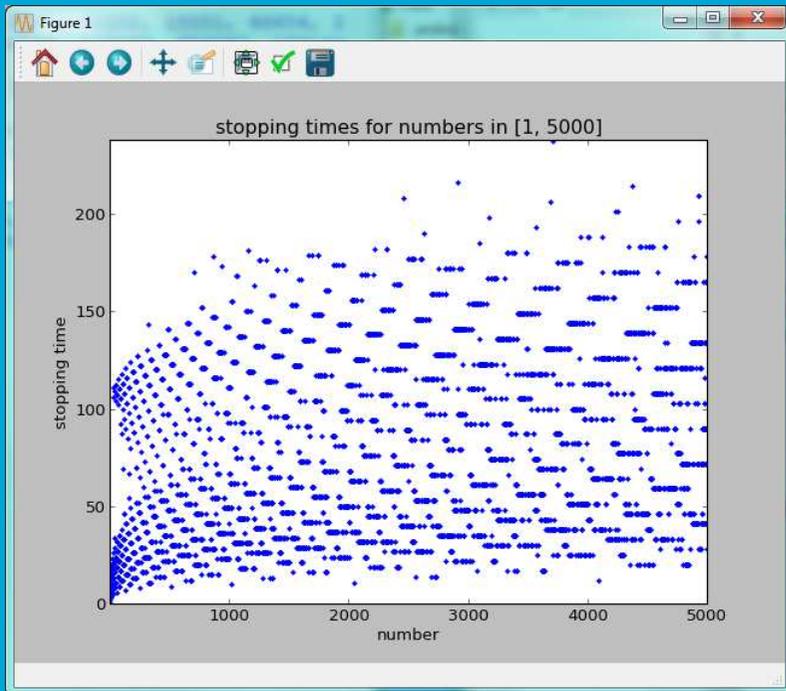
3, 10, 5, 16, 8, 4, 2, 1

Stopping time is 7



Advanced parts of lab assignments

Part 1 and 2



Advanced part 3

Write a program to find the number in $[1, 10^6]$ with the largest stopping time.

Does your program also solve the problem for numbers in $[1, 10^7]$ in acceptable time.

8400511

Stopping time is 685

Examination

- Intermediate test (40% of grade) 3 hours
 - Final exam (60% of grade) 3 hours
-
- Re-exam 3 hours

Example exam question (part a)

An L-system is a parallel rewrite system

Example: rule: $A \rightarrow A+A-A-A+A$
 start with string: 'A'

a) Implement a function Create(n)

Create(0) \rightarrow 'A'

Create(1) \rightarrow 'A+A-A-A+A'

Create(2) \rightarrow

'A+A-A-A+A+A+A+A-A-A+A-A+A-A-A+A-A+A+A+A+A-A-A+A'

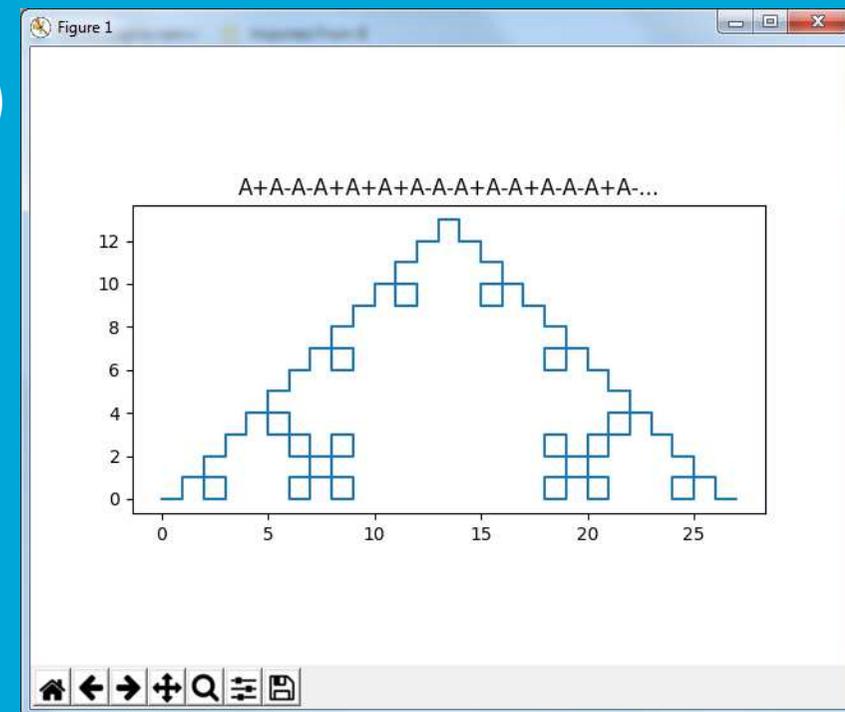
Example exam question (part b)

b) Implement a function Draw(s) (s is a string)

Every character in the string has a meaning:

- A : go forward (one step of length 1)
- + : turn left 90 degrees
- : turn right 90 degrees

s = Create(3)
Draw(s)



Test program

```
#####  
# Below implement the functions create(n) and draw(s)  
#####  
  
##rules:  
##      A -> A+A-A-A-A+A  
##start:  
##      A  
  
def create(n):  
  
    return  
  
def draw(s):  
  
    return
```


Example answer (part a)

```
def create(n):  
  
    s = 'A'  
    for i in range(n):  
        snew = ''  
        for ch in s:  
            if ch == 'A':  
                snew = snew + 'A+A-A-A+A'  
            else:  
                snew = snew + ch  
        s = snew  
  
    return s
```

Example answer (part a; alternative)

```
def create(n):  
  
    s = 'A'  
    for i in range(n):  
        s = s.replace('A', 'A+A-A-A+A')  
  
    return s
```

Example answer (part b)

```
def draw(s):  
    x = [0]  
    y = [0]  
    d = 0  
    for ch in s:  
        if ch == 'A':  
            x.append(x[-1] + cos(d))  
            y.append(y[-1] + sin(d))  
        elif ch == '+':  
            d = d + pi/2  
        elif ch == '-':  
            d = d - pi/2  
  
    ...
```

```
...  
  
t = s[0:30]  
if len(s) > 30:  
    t = t + '...'  
plt.title(t)  
  
plt.plot(x, y)  
plt.axis("scaled")  
plt.show()
```

Results exam and re-exam

Grade	Percentage after exam	Passed after exam	Passed after re-exam
dropout	≈20%		
1.0 – 3.5	≈ 5%		
4.0 – 4.5	≈ 5%		
5.0 – 5.5	≈15%		
6.0 – 6.5	≈10%	≈55%	≈65%
7.0 – 8.5	≈20%		
9.0 – 10.0	≈25%		

Conclusions

- Students learn how to write (small) structured programs
- Students also learn the basics of OO-programming
- Pass rate: about 65% from start population
- Students appreciate interactivity during lectures, but presence is low (30% - 40%)

Questions?