

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Examination in:        INF1100 — Introduction        to  
                                 programming        with        scientific  
                                 applications

Day of examination:    Thursday, October 6, 2014

Examination hours:    15.00 – 19.00.

This examination set consists of 5 pages.

Appendices:            None.

Permitted aids:        None.

Make sure that your copy of the examination set is complete before you start solving the problems.

- Read through the complete exercise set before you start solving the individual exercises. If you miss information in an exercise, you can provide your own reasonable assumptions as long as you explain that in detail.
- The maximum possible score on the exam is 25 points. The maximum number of points is listed for each exercise (a correct answer of a subquestion ((a), (b), etc.) gives 1 point).

### Exercise 1 (10 points)

What is printed in the terminal window when the programs below are run?

(a)

```
a = 4
b = a
a = 2
print 'b:', b
```

*(Continued on page 2.)*

(b)

```
a = 2
for i in range(2):
    a += 2
print a
```

(c)

```
A = [1, 2, 3, -1]
del A[-1]
A.append(-2)
print A
```

(d)

```
import sys
a = 4
b = 6
c = a + b
a = sys.argv[1]
b = sys.argv[2]
print c
```

Execution:

Terminal&gt; python myprog.py 1 2

(e)

```
import numpy as np
x = np.linspace(0, 4, 3)
y = x**2
for x_, y_ in zip(x, y):
    print '%.1f %.1f' % (x_, y_)
```

(f)

```
def P(x):
    return x + 1

def Q(y):
    return 2*y

print Q(P(Q(3)))
```

(g)

*(Continued on page 3.)*

```
import sys
A = [1, 7, 15]

try:
    r = float(A[3])
except IndexError:
    print 'A has length %g' % len(A)
except ValueError:
    print 'Cannot convert %g to float!' % A[3]
```

(h)

```
u = [0, 1]
v = [2, 3]
print u + v
from numpy import array
u = array(u)
v = array(v)
print u + v
```

(i)

```
def plus1(x):
    return x + 1

def test_plus1():
    x = 4.5
    exact = 5.5
    computed = plus1(x)
    tol = 1E-14
    success = abs(exact - computed) < tol
    msg = 'exact=%g, computed=%g' % (exact, computed)
    assert success, msg

test_plus1()
```

(i)

```
for i in range(-1, 1, 2):
    for j in range(3):
        if i == j:
            print i, j
```

(Continued on page 4.)

**Exercise 2 (3 points)**

Write a Python function for computing the sum

$$K(x) = \sum_{k=0}^N (-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

The factorial  $(2k+1)!$  can be computed by `math.factorial(2k+1)`. Print out  $K(\pi)$  for  $N = 4$ .

**Exercise 3 (3 points)**

Extend the program in Exercise 2 such that the  $x$  and  $N$  values are read from the command line. Add a `try-except` block to handle the cases that 1) the user has failed to provide enough command-line arguments or 2) the command-line arguments cannot be interpreted as numbers. Write an error message and stop the program in those cases.

**Exercise 4 (3 points)**

Write the necessary code for plotting the  $K(x)$  function on  $[0, 2\pi]$  in Exercise 2. Mark the  $x$  axis with `x`, the  $y$  axis with `K`, introduce a legend reflecting value of  $N$ , and save the plot to a PNG file.

**Exercise 5 (3 points)**

Write a *test function* for verifying the implementation of the `K(x, N)` function in Exercise 2. Hint: Choose an  $x$  and  $N$ , compute the result by hand, and let the test function compare this exact result with a call to `K(x, N)`. The test function should follow the standard conventions for such functions (i.e., have a name on the form `test_*` and perform the test via an `assert` statement).

**Exercise 6 (3 points)**

A text file with name `simulations.dat` contains a header line, a blank line, and then three columns with numbers:

```
time      measurement  prediction
0.0      -1.376        -1.233
```

(Continued on page 5.)

0.1	-1.091	-1.001
0.2	1.087	0.986
0.3	2.136	3.001

The length of the columns with numbers is not known.

You are asked to read the file in a program, store the data in three Numerical Python (`numpy`) arrays and plot the “measurement” and “prediction” values versus the “time” values in the same figure.

END