UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Examination in:	INF1100 — Introduction to programming with scientific applications
Day of examination:	Monday, October 5, 2015
Examination hours:	15.00 – 19.00

This examination set consists of 6 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).

(Continued on page 2.)

Exercise 1 (10 points)

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

(a)

```
y = 4
y = y*y
print y
```

(b)

```
a = 3
b = a
b = b+a
print a
```

(c)

```
a = 1
for i in range(2):
    a = a*2
print a
```

(d)

```
A = [[-1,0,1],[5,6,7]]
print A[0][-1]
```

(e)

```
import sys
a = sys.argv[1]
b = sys.argv[2]
print eval(a)+eval(b)
```

(Continued on page 3.)

The code is in file `myprog.py`. Execution:

```
Terminal> python myprog.py [0,1] [2,3]
```

(f)

```
dx = 0.25
b = [dx*i for i in range(5)]
print b[-1]
```

(g)

```
from numpy import *
x = linspace(0,1,3)
y = x**2
for x_,y_ in zip(x,y):
    print '%4.2f %4.2f' %(x_, y_)
```

(h)

```
A = ['5', '6', '7', 'end']

try:
    b = float(A[3])
except IndexError:
    print 'A has length %d' %len(A)
except ValueError:
    print 'Cannot convert "%s" to float'% A[3]
```

(i)

```
def f(x):
    return x + 2

def test_f():
    x = 1.0
    expected = 3.0
    computed = f(x)
```

(Continued on page 4.)

```
        tol = 1E-14
        success = abs(exact-computed) < tol
        msg = 'expected %g, computed %g' %(expected,computed)
        assert success, msg
```

(j)

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

def g(y):
    return y**2

x=2
print g(f(g(x)))
```

Exercise 2 (3 points)

A text file with name `densities.dat` contains two header lines and then one column with text and one column with numbers, on the following form:

```
material    density (1000 kg/m^3)
---------------------------------
air         0.0012
gasoline    0.67
ice         0.9
pure water  1.0
seawater    1.025
human body  1.03
limestone   2.6
granite     2.7
iron        7.8
silver      10.5
mercury     13.6
gold        18.9
```

(Continued on page 5.)

```
platinium   21.4
Earth mean  5.52
Earth core  13
Moon        3.3
Sun mean    1.4
Sun core    160
proton      2.3E+14
```

The number of lines in the file is not known. Write a Python program that reads this file, and first stores the result in two lists, one containing the material names and one containing the density values. Then, convert the two lists into a single list, where each item is a pair (list or tuple) containing a material name and corresponding density value.

Exercise 3 (3 points)

We want to write a program that can compute values of the function $f(x) = \sin(a\pi x)$ and its derivative $f'(x) = a\pi\cos(a\pi x)$, where a is some known parameter. Write a Python function `func_deriv(x)` that evaluates and returns the values of $f(x)$ and $f'(x)$. The parameter a can be a global variable. Demonstrate how the function is called, how the returned result can be stored in variables, and how the values of $f(x)$ and $f'(x)$ are written to the screen.

Exercise 4 (3 points)

Extend the program in Exercise 3 so that the parameter a is read from the command line. The function `func_deriv(x)` does not have to be changed. Add a `try-except` block that handles two specific errors; that no command line argument is provided or that it is given in the wrong format. The two errors shall result in different error messages.

(Continued on page 6.)

Exercise 5 (3 points)

Write a function `test_func_deriv()` that tests the function in Exercise 3. The test function should include an `assert` statement. The test should set $a = 1.0, x = 0.25$ and compare the computed values for $f(x)$ and $f'(x)$ to the known analytical values $\sqrt{2}/2$ and $\pi\sqrt{2}/2$. Recall that a tolerance is needed when comparing floating point values.

Exercise 6 (3 points)

The purpose of this exercise is to compute a Taylor polynomial, which is written on the general form

$$p(x) = \sum_{i=0}^{N} t_i(x), \tag{1}$$

and can be used to approximate an arbitrary function. As a specific example, the terms $t_i(x)$ in the Taylor polynomial for $\sin(x)$ are given as

$$t_i(x) = (-1)^i \frac{x^{2i+1}}{(2i+1)!}. \tag{2}$$

Write a function which takes x and the stop value N as input arguments, and returns the sum given by (1) with the individual terms given by (2). The function shall accept an array of arbitrary length for the input argument x, and the return value shall be an array with the same length as x. Remember to include the necessary imports.

Choose x to be an array of 100 uniformly distributed (equally spaced) values in the range $[0, 2\pi]$, and set $N = 3$. Write code for plotting the approximation given by (1) in the same window as the exact function $\sin(x)$.

END