

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Examination in: INF1100 — Introduction to programming with scientific applications

Day of examination: Wednesday, October 10, 2007

Examination hours: 09.00 – 12.00.

This examination set consists of 7 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.
- Your answer must be written on these exercise sheets, and not on separate sheets. In the exercises where you need to write program code we recommend that you first make some notes on a separate sheet of paper before you write down the final solution on these exercise sheets at the marked location.
- Some of the questions are multiple choice exercises. You get points for correct answer. There is only one correct answer in each multiple choice exercise. You will not get points if you do not mark any box or if you mark more than one of the boxes.
- If you have marked one box and afterwards regret your choice, you can write “ERROR” to the left of the box and mark another box.
- The maximum possible score on the exam is 25 points. Every multiple choice question answered correctly results in 1 point, while the programming exercises result in a maximum of 3 points.

*(Continued on page 2.)*

## Multiple choice exercises

### Exercise 1

Which variable name is correct?

- 1var
- \_var
- and
- &top

### Exercise 2

What will be the output of the print statement?

```
C = 21
F = 9*C/5 + 32
print F
```

- 32
- 69.8
- 53
- 69

### Exercise 3

What will be the output of this program?

```
a = 2; b = 5
c = b**a/a
print c
```

- 5
- 13
- 12
- 12.5

*(Continued on page 3.)*

### Exercise 4

For which values of `q` (a `float` or `int`) is the following boolean expression True?

`q < 0 and q > 1`

- All values in `[0,1]`
- None
- All finite values that can be represented on the computer

### Exercise 5

Pick the construction that generates the same list values as the program below:

```
values = []
value = 0.1
stop = 1
while value <= stop:
    values.append(value)
    value += 0.1
```

- `[0.1*i for i in range(10)]`
- `range(0.1, 1.1, 0.1)`
- `[0.1+i for i in range(10)]`
- `[(i+1)*0.1 for i in range(10)]`

### Exercise 6

One of these statements creates the following list. Which one?

`[15, 25, 35, 45, 55]`

- `[i for i in range(15, 55, 10)]`
- `[15, 25] + [45, 55] + [35]`
- `[i for i in range(15, 45, 10)] + [i for i in range(45, 65, 10)]`
- `[5 + 10*i for i in range(5)]`

*(Continued on page 4.)*

### Exercise 7

What will be the output of the print statement?

```
def f(x):  
    return a*x**2  
x = 5; a = 2  
print '%g' % f(x + a)
```

- 50
- 98
- 100
- 49

### Exercise 8

What will be printed by this program?

```
n = 5  
C = []  
for i in range(n):  
    x = i**2  
    C.append(i + x)  
print C
```

- [2, 6, 12, 20, 30]
- [0, 2, 6, 12, 20]
- [0, 2, 6, 12, 20, 30]

### Exercise 9

What is the output of the program below?

```
x = range(3)  
y = [x[1], x[0]+4, x[2] + x[1]]  
xy = [x, y]  
print xy[1][2]
```

- 0
- [0, 1]
- 3

(Continued on page 5.)

## Exercise 10

What will happen when you run the following program:

```
import sys
try:
    t = float(sys[1])
    a = float(sys[2])
except:
    print 'You must provide TWO command-line arguments!!!'
    sys.exit(1)
s = 0.5*a*t**2
print s
```

- The program will correctly compute and print `s` if you provide two numbers as command-line arguments, separated by blank(s)
- The program will always respond with  
You must provide TWO command-line arguments!!!,  
regardless of what you write on the command line
- If you do not provide two numbers as command-line arguments, the program will abort with a `NameError` because there is no `exit` function in the `sys` module

## Exercise 11

How can you turn a list `[1, 2, 3, 4]` into an array of `float` elements?

- `x = array([1, 2, 3, 4], float)`
- `x = array([1, 2, 3, 4])`
- `x = numpy(1, 2, 3, 4)`
- `x = array(1, 2, 3, 4)`

## Exercise 12

What will `x` contain at the end of this program?

```
from numpy import *
x = linspace(0, 10, 3)
s = x
s += 5
```

- `[ 5. 10. 15.]`
- `[ 0. 10. 20.]`
- `[ 0. 3. 6. 9.]`
- `5`
- `[ 0. 5. 10. 5.]`

*(Continued on page 6.)*

### Exercise 13

What is the correct `numpy` syntax for allocating a one-dimensional array (or vector) with four elements?

- `a = zeros(4)`
- `a = eye(4)`
- `a = array(4)`
- `a = linspace(4)`

## Programming exercises

### Exercise 14

Make a Python function `g(t, a)` for computing  $g(t) = e^{-at^2}$  and  $g'(t) = -2atg(t)$ . It should return the function values of  $g(t)$  and  $g'(t)$ .

```
def g(t, a):
    g_t = exp(-a*t**2)
    dg_t = -2*a*t*g_t
    return g_t, dg_t
```

### Exercise 15

An arbitrary triangle can be described by the coordinates of its three vertices:  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . The area of the triangle is given by the formula

$$A = \frac{1}{2} [x_2y_3 - x_3y_2 - x_1y_3 + x_3y_1 + x_1y_2 - x_2y_1] .$$

Write a function `area(vertices)` that returns the area of a triangle whose vertices are specified by the argument `vertices`, which is a nested list of the vertex coordinates. For example, `vertices` can be `[[0,0], [1,0], [0,2]]` if the three corners of the triangle have coordinates  $(0,0)$ ,  $(1,0)$  and  $(0,2)$ .

```
def area(vertices):
    x1 = vertices[0][0]; y1 = vertices[0][1]
    x2 = vertices[1][0]; y2 = vertices[1][1]
    x3 = vertices[2][0]; y3 = vertices[2][1]
    A = 0.5*(x2*y3 - x3*y2 - x1*y3 + x3*y1 + x1*y2 - x2*y1)
    return A
```

(Continued on page 7.)

**Exercise 16**

The function

$$f(x, t) = e^{-(x-3t)^2} \sin(3\pi(x-t))$$

describes, for a fixed value of  $t$ , a wave localized in space. Make a program that plots this function as a function of  $x$  on the interval  $[-4, 4]$  when  $t = 0$ .

```

from scitools.all import *
def f(x, t):
    return exp(-(x - 3*t)**2)*sin(3*pi*(x - t))
t = 0
x = linspace(-4, 4, 101)
y = f(x, t)
plot(x, y)

```

**Exercise 17**

A loan  $L$  is to be paid back with a fixed amount  $L/N$  every month over  $N$  months. The annual interest rate for the loan is  $p$  percent, so the monthly interest rate is  $\frac{p}{12}$ . The size of the loan after  $n$  months is  $x_n$ , and  $x_n$  changes from one month to the other according to

$$x_n = x_{n-1} + \frac{p}{12 \cdot 100} x_{n-1} - \frac{L}{N}, \quad x_0 = L, \quad i = 1, \dots, N.$$

Make a program that solves this equation.

```

L = 10000
p = 5.
N = 36
x0 = L
xnm1 = x0
n = 1
while n <= N:
    xn = (1 + p/1200.)*xnm1 - L/float(N)
    xnm1 = xn
    n += 1
print xn

```

END