

Aarhus Universitet, Science and Technology, Computer Science

Exam

Introduction to Programming with Scientific Applications

Wednesday 27 June 2018, 9:00-11:00

Allowed aid: **None**

The exam questions are answered on the problem statement that is handed in at the end of the exam

*Tilladte hjælpemidler: **Ingen***

Eksamensspørgsmålene besvares på opgaveformuleringen, som afleveres ved eksamenens slutning

Student ID/*årskort*

Name/*navn*

Information

The exam consists of a set of multiple-choice questions. The questions are answered on the problem statement **that is handed in**. For each question is stated the weight of the question compared to the full exam. Each sub-question has exactly one correct answer. You can select **at most one** answer for each sub-question, by marking the corresponding box with a cross. A sub-question is scored as follows:

- Marking the correct answer gives you 1 point.
- If you do not mark any answer you get 0 points.
- Marking a wrong answer gives you $-\frac{1}{k-1}$ point, where k is the number of answer options.

For a question with weight $v\%$ containing n sub-questions, where you score a total of s points, your score for the question will be $\frac{s}{n} \cdot v\%$. Note that it is possible to get a negative score for a question.

*Dette eksamenssæt består af en mængde multiple-choice-opgaver. Opgaverne besvares på opgaveformuleringen **som afleveres**. For hver opgave er angivet opgavens andel af det samlede eksamenssæt. Hvert delspørgsmål har præcist et rigtigt svar. For hvert delspørgsmål, må man vælge **max ét svar** ved at afkrydse den tilsvarende rubrik. Et delspørgsmål bedømmes som følgende:*

- *Hvis du sætter kryds ved det rigtige svar, får du 1 point.*
- *Hvis du ikke sætter nogen krydser, får du 0 point.*
- *Hvis du sætter kryds ved et forkert svar, får du $-\frac{1}{k-1}$ point, hvor k er antal svarmuligheder.*

For en opgave med vægt $v\%$ og med n delspørgsmål, hvor man opnår samlet s point, beregnes pointene for besvarelse af opgaven som $\frac{s}{n} \cdot v\%$. Bemærk at det er muligt at få negative point for en opgave.

Python version

In the following Python refers to Python 3.6.

I det følgende antages at Python refererer til Python 3.6.

Question 4 (Boolean operations, 5 %)

What is the result of each of the below boolean expressions?

Hvad er resultatet af hvert af nedenstående boolske udtryk?

	False	True	"False"	"True"	None
False and True					
None or True					
"False" and "True"					
"False" or "True"					
"False" and not "True"					

Question 5 (Conditionals, 4 %)

```
x = 1
if x == 1:
    x += 2
    if x == 1:
        x += 3
    else:
        x += 2
else:
    x += 1
print(x)
```

What does the above code print?

1 2 3 4 5 6 7 8 9

Hvad udskriver ovenstående kode?

Question 6 (Loop, 4 %)

```
x = 1
while True:
    if x > 3:
        break
    x += 1
else:
    1
print(x)
```

What does the above code print?

1 2 3 4 5 6 7 8 9

Hvad udskriver ovenstående kode?

Question 7 (Function, 4 %)

```
def f(x, y):  
    return x ** y  
  
x = 2  
y = 3  
print(f(y, x))
```

What does the above code print?

2 3 4 6 8 9

Hvad udskriver ovenstående kode?

Question 8 (Recursion, 4 %)

```
def f(x):  
    if x >= 1:  
        f(x - 1)  
        print(x, end=' ')  
        f(x - 1)  
  
f(3)
```

What does f(3) print?

3211211 1121123 1213121 12321 232

Hvad udskriver f(3)?

Question 9 (List comprehension, 5 %)

What is the result of each of the below expressions?

Hvad er resultatet af hvert af nedenstående udtryk?

1 2 3 4 5 6 7 8 9 10

```
sum([x*x for x in [1,2,1]])  
sum([y for x in (1, 2, 3) for y in (1, 2)])  
sum([x*y for x,y in zip([1,2,3],[3,1,1])])  
sum([y for x,y in [(1,2),(3,4),(2,1)])])  
sum([x*x for x in [1,2,3] if x < 3])
```

Question 10 (sorted, 4 %)

```
print(sorted(("pab","kd","defg","s"), key=len, reverse=True))
```

What does the above program print?

Hvad udskriver ovenstående program?

```
['defg', 'kd', 'pab', 's']  
['s', 'pab', 'kd', 'defg']  
['s', 'kd', 'pab', 'defg']  
['defg', 'pab', 'kd', 's']  
['pab', 'defg', 'kd', 's']
```

Question 11 (map, 4 %)

```
print(tuple(map(lambda x : x + x, range(3))))
```

What does the above program print?

Hvad udskriver ovenstående program?

```
(0, 1, 2) (3) (0, 1, 3) (0, 1, 2 ,3) (0, 2, 4) (0, 2, 4, 9)
```

Question 12 (Types, 5 %)

What is the resulting type of each of the below expression?

For hvert af nedenstående udtryk hvilken type har resultatet?

	int	float	str	list	tuple	bool
<code>1 + 2.0</code>						
<code>3 / 2</code>						
<code>3 // 2</code>						
<code>'abc' + 'def'</code>						
<code>'abc' * 3</code>						
<code>len(('a', 'b'))</code>						

Question 13 (yield, 4 %)

```
def f(x):  
    yield 1  
    while x > 0:  
        yield x  
        x -= 1  
    yield 1  
  
print(list(f(3)))
```

What list does the above program print?

Hvilken liste udskriver ovenstående program?

[1] [1,1] [1,3,2,1,1] [1,3,3,3,1] [1,3,2,1,0,1] [1,3,2,1]

Question 14 (Recursive data, 4 %)

```
def traverse(T):  
    if type(T) is not tuple:  
        return T + T  
    else:  
        return tuple(traverse(c) for c in T)  
  
print(traverse(((3,(2,5)),4)))
```

What does the above program print?

Hvad udskriver ovenstående program?

((3, 3, (2, 2, 5, 5)), 4, 4)
(((3, 3), ((2, 2), (5, 5))), (4, 4))
(3, 2, 5, 4)
(6, 4, 10, 8)
((6, (4, 10)), 8)
(3, 3, 2, 2, 5, 5, 4, 4)

Question 15 (Generator expressions, 4 %)

```
g = (x * 2 for x in range(1, 6))  
  
print(next(g), next(g), list(g))
```

What does the above program print?

Hvad udskriver ovenstående program?

- [2, 4, 6, 8, 10]
- [2, 4, 6, 8, 10]
- [6, 8, 10]
- [6, 8, 10]
- None []

Question 16 (* notation, 4 %)

```
x, y, *z, w = 1, 2, 3, 4, 5, 6
```

What is the value of z?

Hvilken værdi har z?

- [3]
- [3,4,5]
- [3,4,5,6]
- 12

Question 17 (Sets, 5 %)

What is the value of the following expressions?

Hvilken værdi har følgende udsagn?

- []
 - [1]
 - [2]
 - [3]
 - [2,2]
 - [1,2,3]
 - [1,2,2,3]
- ```
list({1, 2} | {2, 3})
list({1, 2} & {2, 3})
list({1, 2} - {2, 3})
list({1, 2} - {1, 2})
list({1, 2} - {1, 2, 3})
```



### Question 18 (zip, 4 %)

---

```
list(zip([1,2], zip([3,4], [5,6])))
```

---

What is the value of the above expressions?

*Hvilken værdi har ovenstående udsagn?*

[(1, 2), (3, 4), (5,6)]  
[1, 3, 5, 2, 4, 6]  
(1, 3, 5), (2, 4, 6)  
[(1, (3, 5)), (2, (4, 6))]  
[((1, 3), 5), ((2, 4), 6)]

### Question 19 (numpy, 4 %)

---

```
import numpy as np
print(np.array([1,2,3]) + np.array([4,5,6]))
print(np.array([1,2,3]) * np.array([4,5,6]))
print(3 * np.array([1,2,3]))
print(np.array([1,2,3]) @ np.array([4,5,6]))
```

---

What does the above program print?

*Hvad udskriver ovenstående program?*

[1 2 3 4 5 6]  
32  
[1 2 3 1 2 3 1 3 3]  
4  
  
[1 2 3 4 5 6]  
[ 4 10 18]  
[1 2 3 1 2 3 1 3 3]  
4  
  
[5 7 9]  
[ 4 10 18]  
[3 6 9]  
32  
  
[1 2 3 4 5 6]  
[ 4 10 18]  
[3 6 9]  
32

### Question 20 (lambda, 4 %)

---

```
def f(g, h, L):
 return [g(x) if x % 2 == 0 else h(x) for x in L]

print(f(lambda x : 3 * x, lambda x : 2 * x, [1, 2, 3, 5]))
```

---

What list does the above program print?

*Hvilken liste udskriver ovenstående program?*

[1,0,1,1] [3,6,9,10] [3,4,9,15] [2,6,6,10] [2, 6, 6, 15]

### Question 21 (Assertions, 4 %)

---

```
def f(x):
 assert type(x) == int

 return 3 * x
```

---

What does each of the below function calls return?

*Hvad returnerer hvert af nedenstående funktionskald?*

6 6.0 '222' AssertionError

---

f(2)

f(2.0)

f('2')

## Question 22 (Inheritance, 5 %)

---

```
class A:
 def say(self):
 print("Hello")

class B:
 def greet(self):
 print("Welcome")

class C(A, B):
 pass
```

---

What does each of the below metode calls print?

*Hvad udskriver hvert af nedenstående metodekald?*

\_\_\_\_\_  
Hello   Welcome   AttributeError

A().say()

B().say()

C().say()

A().greet()

B().greet()

C().greet()

---

### Question 23 (Decorator, 4%)

---

```
def d(x):
 def y(z):
 print(type(x))
 x(z+z)

 return y

@d
def f(x):
 print("value = " + str(x))

f(21)
```

---

What does the above program print?

*Hvad udskriver ovenstående program?*

```
<class 'int'>
value = 21

<class 'int'>
value = 42

<class 'function'>
value = 21

<class 'function'>
value = 42

<class 'str'>
value = 21

<class 'str'>
value = 42
```