Aarhus Universitet, Science and Technology, Computer Science

Mock 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 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 1 (Basic types, 4 %)

Betragt udførslen af følgende Python kode,

```
x = input()
y = input()
```

hvor brugeren giver følgende input

```
2
3
```

Hvad svarer Python når vi efterfølgende udfører hver af nedenstående linier?

	5	6	23	33	222	'23'	'33'	'222'	TypeError
`x*y`									
`int(x)*int(y)`									
`x*int(y)`									
`x+y`									
`x+int(y)`									
`int(x)+int(y)`									

Question 2 (List comprehension, 4 %)

Assume we want to generate a list of the numbers 1^2, 2^2, 3^2, ..., 10^2. For each of the below expression state if the expression generates this list.

Antag vi vil genere en liste med tallene 1^2, 2^2, 3^2, ..., 10^2. *For hvert af nedenstående udtryk angiv om udtrykket generer denne liste.*

	Yes/*Ja*	No/*Nej*
`[x ** 2 in range(10)]`		
`[x ** 2 for x in range(10)]`		
`[x ** 2 for x in range(1, 11)]`		
`[x * y for x, y in enumerate(range(1, 11), start=1)]`		

Question 3 (Types, 4 %)

What is the type of the following expressions?

Hvad er typen af følgende udtryk?

	List	Tuple	Dictionary	Set	Int
`{}`					
`{42}`					
`{42, 42}`					
`{42: 42}`					
`()`					
`(42)`					
`(42,)`					
`(42, 42)`					
`[42]`					
`[42, 42]`					

Question 4 (Mutability, 4 %)

State for each of the below Python types if values of the type are mutable or not.

Angiv for hver af nedenstående Python typer om værdier af typerne er muterbare eller ej.

	Mutable	Immutable
Integers (**`int`**)		
Strings (**`str`**)		
Floating point numbers (**`float`**)		
Lists (**`list`**)		
Tuples (**`tuple`**)		
Sets (**`set`**)		
Dictionaries (**`dict`**)		
Boolean (**`bool`**)		

Question 5 (Lists, 4 %)

```python
x = ['a', 'b']
y = x
z = x + ['c']
y.append('c')
```

What is the value of `x`, `y` and `z` after executing the above code?

Hvad er værdien af `x`*,* `y` *og* `z` *efter udførslen af ovenstående kode?*

	['a', 'b']	['a', 'b', 'c']	['a', 'b', 'c', 'c']
x			
y			
z			

Question 6 (Boolean, 4 %)

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

Hvad er resultat af hvert af nedenstående boolske udtryk?

	True	False	ZeroDivisionError
`1 == '1' and 1/0 == 0`			
`1 == '1' or 1/0 == 0`			
`False or not False`			

Question 7 (Recursion, 4 %)

```python
def rec_print(x):
    print(x, end=' ')
    if x > 1:
        rec_print(x - 1)
        rec_print(x - 1)
```

What does `rec_print(3)` print?

Hvad udskriver `rec_print(3)`*?*

```
3 2 2     3 2 2 1 1     3 2 1 2 1     3 2 1 1 2 1 1     3 2 2 1 1 1 1
```

Question 8 (`for` loops, 4 %)

```
for x in range(3):
    for y in range(2):
        print(x, y, sep='', end='.')
```

What does the above code print?

Hvad udskriver ovenstående kode?

`11.21.31.12.22.32.` `11.12.21.22.31.32.` `00.01.10.11.20.21.` `00.10.20.01.11.21.`

Question 9 (Keyword arguments, 4 %)

```
def f(x, y=3, z=5):
    print(x + y + z)
```

For each of the below function calls state the value is returned or if the call is illegal.

Angiv for hvert af nedenstående funktionskald returværdien eller om kaldet er ulovligt.

	5	6	7	8	9	10	TypeError
`f()`							
`f(1)`							
`f(1, 2)`							
`f(1, x=2)`							
`f(1, y=2)`							
`f(1, z=2)`							
`f(1, 2, 3)`							
`f(1, 2, z=3)`							
`f(1, 2, y=3)`							

Question 10 (zip, 4 %)

What does `list(zip((1, 2), (3, 4)))` return?

Hvad returnerer `list(zip((1, 2), (3, 4)))`*?*

```
[((1,2),(3,4))]   [(1,3),(2,4)]   [1,2,3,4]   [1,3,2,4]
```

Question 11 (Conditional expressions, 4 %)

```
print(sum([2 * x if x % 2 == 0 else 3 * x for x in [2, 3]]))
```

What does the above program print?

Hvad udskriver ovenstående program?

```
2    3    5    4    6    9    12    13    15
```

Question 12 (`sorted`, 4 %)

```
print(sorted(["abx", "CDK", "Ghy"],
      key=lambda x: sorted(x.lower(), reverse=True)))
```

What does the above program print?

Hvad udskriver ovenstående program?

```
["abx", "CDK", "Ghy"]
["CDK", "abx", "Ghy"]
["CDK", "Ghy", "abx"]
["abx", "Ghy", "CDK"]
["Ghy", "abx", "CDK"]
["Ghy", "CDK", "abx"]
```

Question 13 (`matplotlib`, 4 %)

```
import matplotlib.pyplot as plt
plt.plot([0, 1], [2, 3], ".")
plt.show()
```

What are the two points plotted by the above program?

Hvilke to punkter plottes med ovenstående program?

```
(0,1)  (0,1)  (0,2)  (0,2)
(2,3)  (2,4)  (1,3)  (1,5)
```

Question 14 (List arithmetic, 4 %)

```
print(3 * [1, 2, 3])
```

What does the above program print?

Hvad udskriver ovenstående program?

```
[3,2,3]  [3,6,9]  [1,2,3,1,2,3,1,2,3]  [1,1,1,2,2,2,3,3,3]  TypeError
```

Question 15 (`lambda`, 4 %)

```
def square(x):
    return x * x

g = lambda f: (lambda x: f(f(x)))

print(g(square)(3))
```

What does the above program print?

Hvad udskriver ovenstående program?

```
3   6   9   18   81
```

Question 16 (Decorator, 4 %)

```
def f(g):
    def h(x):
        return g(g(x))
    return h

@f
def p(s):
    print(s)
    return s + s

p("say!")
```

What does the above program print?

Hvad udskriver ovenstående program?

```
say!
```

```
say!
say!
```

```
say!
say!say!
```

```
say!say!
```

```
say!say!
say!say!
```

Question 17 (Characters, 4 %)

```
def f(s):
    a = ord('a')
    return ''.join(chr(((ord(c) - a) + 1) % 26 + a) for c in s)

print(f('zebra'))
```

What does the above program print?

Hvad udskriver ovenstående program?

`zebra` `ebraz` `azebr` `afcsb` `ydaqz`

Question 18 (Generator, 4 %)

```
def gen(s):
    reported = set()
    for x in s:
        if x not in reported:
            reported |= set(x)
            yield x

print(list(gen('hello_world'))[0:7])
```

What does the above program print?

Hvad udskriver ovenstående program?

```
'helo_wr'
['helo_wr']
['h', 'e', 'l', 'l', 'o', '_', 'w']
['h', 'e', 'l', 'o', '_', 'w', 'r']
[{'h'}, {'e'}, {'l'}, {'o'}, {'_'}, {'w'}, {'o'}]
```

Question 19 (Class, 4 %)

```
class C:
    def __init__(self, x=1, y=2):
        self.x = x
        self.y = y

    def sum(self):
        return self.x + self.y

print(C(y=3).sum())
```

What does the above program print?

Hvad udskriver ovenstående program?

```
1    2    3    4    5    TypeError
```

Question 20 (Recursive generator, 4 %)

```
def f(x):
    if type(x) is not tuple:
        yield x
    else:
        for c in x:
            for z in f(c):
                yield z

print(list(f((((1, 2), 3), (4, 5)))))
```

What does the above program print?

Hvad udskriver ovenstående program?

```
[1]   [3,5]   [[[1,2],3],[4,5]]   [1,2,3,4,5]   [5]
```

Question 21 (Method calls, 4 %)

```
class A:
    name = "My name"

    def __init__(self, s):
        self.name = s

a = A('Hugo')
print(a.name)
print(A.name)
```

What does the above program print?

Hvad udskriver ovenstående program?

```
Hugo          Hugo          My name          Hugo
Hugo          My name       My name          AttributeError
```

Question 22 (Attributes, 4 %)

```
class A:
    name = "My name"

    def __init__(self, s):
        self.name = s

a = A('Hugo')
```

Which of the below terms apply to `A.name`?

Hvilke af nedenstående beskrivelser er korrekt for `A.name`*?*

	Yes/*Ja*	No/*Nej*
attribute		
class attribute		
data attribute		
method attribute		

Which of the below terms apply to `a.name`?

Hvilke af nedenstående beskrivelser er korrekt for `a.name`*?*

	Yes/*Ja*	No/*Nej*
attribute		
class attribute		
data attribute		
method attribute		

Which of the below terms apply to `A.__init__`?

Hvilke af nedenstående beskrivelser er korrekt for `A.__init__`*?*

	Yes/*Ja*	No/*Nej*
attribute		
class attribute		
data attribute		
method attribute		

Question 23 (super, 4 %)

```
class A:
    def say(self):
        print("Hi there")

class B(A):
    def say(self):
        print("Howdy")

    def __init__(self):
        self.say()
        super().say()

A()
B()
```

What does the above program print?

Hvad udskriver ovenstående program?

```
Hi there     Hi there     Hi there     Hi there     Howdy
Hi there     Hi there     Howdy        Howdy        Hi there
Hi there     Howdy        Hi there
```

Question 24 (Precedence, 4 %)

```
print(2 + 2 * 2 ** 2)
```

What does the above program print?

Hvad udskriver ovenstående program?

```
8    10    16    18    64
```

Question 25 (Decorators, 4 %)

```
import os

class Logger:
    logs = []
    def __init__(self):
        pass
    def log(self, key, value):
        self.logs.append((key, value))
    def print(self):
        for k,v in self.logs:
            print(k, ":" , v)

def log_call(logger):
    def decorator(func):
        def wrapper(*args, **kwargs):
            logger.log("call", func.__name__)
            return func(*args, **kwargs)
        wrapper.__name__ = func.__name__
        return wrapper
    return decorator

def log_output(logger):
    def decorator(func):
        def wrapper(*args, **kwargs):
            s = func(*args, **kwargs)
            logger.log("output(" + func.__name__ + ")", s)
            return s
        wrapper.__name__ = func.__name__
        return wrapper

    return decorator

mlog = Logger()

@log_output(mlog)
@log_call(mlog)
def test(string):
    return string

test("This is my personal string")
mlog.print()
```

What does the above program print?

Hvad udskriver ovenstående program?

```
output(test) :  This is my personal string
call :  test
```

```
call :  test
output(test) :  This is my personal string
```

```
test
This is my personal string
```

```
call :  test
output(wrapper) :  This is my personal string
```

```
call :  wrapper
output(wrapper) :  This is my personal string
```