MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Beräkningsmatematik
Examinator: Lars Arvestad

Tentamensskrivning i
DA2005 Programmeringsteknik *och*
DA2004 Programmeringsteknik för matematiker
7.5 hp
2023-10-03

**Please note: this exam contains the questions in Swedish *and* English!**

- **Skriv tydligt.** Svårlästa svar riskerar 0 poäng.
- Skriv bara på en sida av varje papper!
- Motivera alla svar (om inte annat anges)!
- Man måste bli godkänd på del A (5 rätt på 10 frågor) för att del B ska rättas.
- **Hjälpmedel:** Ett A4 med så mycket information du vill. Du får skriva på båda sidorna.
- **Betygsgränser:** E: 10, D: 12, C: 14, B: 16, A: 18, av maximala 20.

---

## Del A: flervalsfrågor

*Var snäll samla svaren på del A på ett svarspapper.*

**1**. Vad blir värdet på variabeln `x` i koden till höger?

A. 3

B. 5

C. 7

D. 9

E. Inget då ett särfall lyfts.

```
lst = [[1], [[1,2], [3,4], []], (6,9,2)]
x = sum([len(item) for item in lst])
```

**2**. Vad skrivs ut av koden till höger?

A. Särfall lyfts då det inte är tillåtet att blanda **`for`** och **`while`**.

B. `[(0, 'a'), (0, 'b')]`

C. `[(1, 'a'), (1, 'b')]`

D. `[(1, 'a'), (1, 'b'), (0, 'a'), (0, 'b')]`

E. `[(2, 'a'), (2, 'b'), (1, 'a'), (1, 'b')]`

```
i = 1
lst = []
while i > 0:
    for y in ['a', 'b']:
        lst.append((i, y))
    i -= 1
print(lst)
```

**3**. Om funktionen `fkn` är definierad som till höger, vad returnerar anropet `fkn(4)`?

A. 2

B. 3

C. 4

D. 5

E. 6

```
def fkn(n):
    a = 0
    b = 1
    for i in range(n):
        c = a + b
        a = b
        b = c
    return b
```

**4**. Vilket alternativ är en korrekt beskrivning av begreppet *metod* inom objektorientering?

A. Det är som en funktion, fast begränsad till objekt från en eller flera klasser.
B. Det är den funktion som används för att instantiera klassens objekt.
C. En metod används så att en klass kan ärva egenskaper från en annan klass.
D. Objektorienterade algoritmer kallas metoder.
E. Den samlade beskrivningen av en klass kallas metod.

**5**. Om man exekverat `from math import sqrt`, hur anropar man `sqrt` för att beräkna roten ur `x`? Ett svar är rätt.

A. `sqrt(x)`
B. `sqrt x`
C. `math.sqrt(x)`
D. `math.sqrt x`
E. `module.math.sqrt(x)`

**6**. Hur skriver man en dokumentationssträng (*docstring*) för en funktion eller metod?

A. Man lägger en sträng efter `def`-raden.
B. Man lägger en sträng på raden innan `def`.
C. Man lägger en kommentar (raden börjar med `#`) efter `def`-raden.
D. Man lägger en kommentar (raden börjar med `#`) på raden innan `def`.

**7**. Vilken eller vilka boolska variabeltilldelningar gör att utrycket

```
(x and y) or (x and z)
```

evaluerar till `True`?

A. `x=False, y=False, z=True`
B. `x=False, y=True, z=False`
C. `x=True, y=False, z=True`
D. `x=True, y=True, z=False`
E. `x=True, y=True, z=True`

**8**. Vilka påståenden är sanna?

A. Funktioner kan returnera funktioner.
B. Funktioner kan inte definieras utan parametrar.
C. Funktioner kan anropa sig själva.
D. Funktioner kan ha flera **return**-satser.
E. Inget av ovanstående.

**9**. Vad blir det för utskrift av koden till höger?

A. 2
B. 3
C. 4
D. 5
E. Ingenting, det blir ett särfall.

```
def f(x):
    return x*y

def g(y):
    return x*y

x = 1
y = 2
print(f(2) + g(1))
```

**10**. Hur skapar man en modul *m* i Python, så att man i sitt program kan skriva `import m`?

A. Skapa filen m.py och använd nyckelordet *module* för att markera att det är en modul.
B. Skapa filen m.py och definiera en klass i filen.
C. Skapa filen m.py och definiera en klass i filen. Klassen ska ärva från basklassen "module".
D. Skapa filen m.py och skriv högst upp `export f1, f2` osv, där f1 och f2 är funktioner i m.py som ska exporteras.
E. Skapa filen m.py och gör ingenting: alla Python-filer kan användas som om de vore moduler.

## Del B: kodfrågor

*Var snäll använd ett papper till varje fråga i del B.*

**11**. I Python-kod kan man ibland se rader som denna:

```
assert x > 0
```

Vad innebär det? (1p)

**12**. I koden nedan finns en funktion som skriver ut en uppslagstabell med par av strängar. Utskriften ska anpassas så att båda kolumnerna är vänsterjusterade. Exempelkörningen visar hur utdata är *tänkt* att bli. Tyvärr finns det minst två fel i koden. Vilka? (2p)

```
def tabulate_dict(d):
    '''
    Tabulate the dictionary d, assumed to contain strings as keys and values.
    The right column is properly left indented.
    '''
    for key, val in d.items():
        if len(key) > left_len:
            left_len = len(key)

    for key, val in d.items():
        n_spaces = len(key)
        spacer = ' ' * n_spaces
        print(key, spacer, val)
```

**Exempelkörning, vid korrekt kod:**

```
[In: ] butterflies = {
    'Sorgmantel': 'Nymphalis antiopa',
    'Nässelfjäril': 'Aglais urticae',
    'Grönsnabbvinge': 'Callophrys rubi',
    'Amiral': 'Vanessa atalanta',
    'Citronfjäril': 'Gonepteryx rhamni',
    }
[In: ] tabulate_dict(butterflies)
[Out:]
Sorgmantel      Nymphalis antiopa
Nässelfjäril    Aglais urticae
Grönsnabbvinge  Callophrys rubi
Amiral          Vanessa atalanta
Citronfjäril    Gonepteryx rhamni
```

**13**. Skriv funktionen `count_swedish_alphabet(s)` som räknar och returnerar antalet bokstäver i det svenska alfabetet i s. (1p)

**Exempelkörning:**

```
[In: ] count_swedish_alphabet('')
[Out:] 0
[In: ] count_swedish_alphabet('ääö')
[Out:] 3
[In: ] count_swedish_alphabet('abc')
[Out:] 3
[In: ] count_swedish_alphabet('AbCdefghijklmnopqrstuvwxyzÅÄÖ)
[Out:] 29
[In: ] count_swedish_alphabet('_(}01:')
[Out:] 0
```

**14**. I den här uppgiften ska du skriva en funktion som räknar tomma rader. Om du vill kan du svara med en funktion som löser båda uppgifterna samtidigt.

A. Skriv funktionen `count_empty_lines(filename)` som returnerar antalet tomrader i filen som ges av `filename`. En tomrad definieras som att det bara finns ett ensamt nyradstecken. Om filen inte har något innehåll alls så finns det noll tomrader. En fil med endast ett nyradstecken har en tomrad. (1p)

B. Utöka funktionen `count_empty_lines(filename)` till att hantera problem. Om filen inte finns eller inte kan läsas så ska en varning skrivas ut (”Warning: could not read file”) och 0 returneras. (1p)

**15**. Skriv en funktion `compose(f, g)` som returnerar en funktion som beräknar `f(g(x))`. Du ska utgå ifrån att både `f` och `g` tar exakt ett argument. (1p)

**Exempelkörning:**

```
[In: ] def double(x):
           return 2*x
[In: ] def triple(x):
           return 3*x
[In: ] h = compose(double, triple)
[In: ] h(1)
[Out:] 6
[In: ] h(0)
[Out:] 0
[In: ] h(2)
[Out:] 12
```

**16**. Skriv klassen `Polynomial` för att representera polynom. I laboration 2 skrev ni kod för att arbeta med polynom som listor och nu ska du skriva en klass för att skapa bättre abstraktion.

I din klass ska `multiply` definieras med en konstant faktor som argument, så att man kan skala koefficienterna i polynomet med faktorn. Du ska också ha `__str__` definierad så att utskrifterna blir bra. Se exempelkörningen nedan för hur de ska fungera!

Du kan använda funktionen `poly_to_string(p_list)` från laboration 2 för att konvertera en lista av koefficienter till en sträng som beskriver polynomet. Dvs, du behöver inte skriva den utan kan anropa den som om den fanns i koden. (3p)

**Exempelkörning:**

```
[In: ] p1 = Polynomial([3, 2, 1])
[In: ] print(p1)
[Out:] 3 + 2x + x^2
[In: ] p2 = Polynomial([1, 1, 1])
[In: ] print(p2)
[Out:] 1 + x + x^2
```

```
[In: ] p2.multiply(7)
[In: ] print(p2)
[Out:] 7 + 7x + 7x^2
```

# English translation

- This exam has multiple choice questions where at least one answer is correct. If your answer is incorrect or you do not include all correct answers, you will receive 0 points on that question.
- **Write clearly.** Answers that are difficult to read may receive 0 points.
- Write only on one side of each paper!
- You must pass part A (5 correct out of 10 questions) to have your part B graded.
- **Aids:** An A4 with as much information as you want. You can write on both sides.
- **Grade thresholds:** E: 10, D: 12, C: 14, B: 16, A: 18, of maximum 20.

---

## Part A: Multiple choice

*Please collect your answers to part A on a single piece of paper.*

**1**. What value will the variable `x` get in the code on the right?

A. 3

B. 5

C. 7

D. 9

E. None, an exception is raised.

```
lst = [[1], [[1,2], [3,4], []], (6,9,2)]
x = sum([len(item) for item in lst])
```

**2**. What is printed by the code on the right?

A. An exception is raised because you cannot mix **`for`** and **`while`**.

B. `[(0, 'a'), (0, 'b')]`

C. `[(1, 'a'), (1, 'b')]`

D. `[(1, 'a'), (1, 'b'), (0, 'a'), (0, 'b')]`

E. `[(2, 'a'), (2, 'b'), (1, 'a'), (1, 'b')]`

```
i = 1
lst = []
while i > 0:
    for y in ['a', 'b']:
        lst.append((i, y))
    i -= 1
print(lst)
```

**3**. If the function `fkn` is defined as on the right, what will the call `fkn(4)` return?

A. 2

B. 3

C. 4

D. 5

E. 6

```
def fkn(n):
    a = 0
    b = 1
    for i in range(n):
        c = a + b
        a = b
        b = c
    return b
```

**4**. Which alternative is a correct description of the concept *method* in object-oriented programming?

A. It is like a function, but limited to the objects from one or more classes.

B. It is a function which is used to instantiate objects for the class.

C. A method is applied to inherit properties from another class.

D. Object-oriented algorithms are called methods.

E. The code describing a class is called a method.

**5**. If you have evaluated `from math import sqrt`, how do you call `sqrt` for computing the square root of `x`? One answer is correct.

A. `sqrt(x)`
B. `sqrt x`
C. `math.sqrt(x)`
D. `math.sqrt x`
E. `module.math.sqrt(x)`

**6**. How do you write a docstring for a function or method?

A. You put a string after the `def` line.
B. You put a string on the line prior to `def`.
C. You put a comment (line starting with `#`) after the `def` line.
D. You put a comment (line starting with `#`) on the line prior to `def`.

**7**. Which boolean variable assignment or assignments makes the expression

```
(x and y) or (x and z)
```

evaluate to `True`?

A. `x=False, y=False, z=True`
B. `x=False, y=True, z=False`
C. `x=True, y=False, z=True`
D. `x=True, y=True, z=False`
E. `x=True, y=True, z=True`

**8**. Which claims are true?

A. Functions can return functions.
B. Functions cannot be defined without parameters.
C. Functions can call themselves.
D. Functions can have several **`return`** statements.
E. None of the above.

**9**. What is printed by the code on the right?

A. 2
B. 3
C. 4
D. 5
E. Nothing, there will be an exception.

```
def f(x):
    return x*y

def g(y):
    return x*y

x = 1
y = 2
print(f(2) + g(1))
```

**10**. How do you create a module in Python, so that you can write `import m` in your program?

A. Create the file m.py and use the keyword *module* to indicate that it is a module.
B. Create the file m.py and define a class in it.
C. Create the file m.py and define a class in it. The class must inherit from the base class "module".
D. Create the file m.py and write `export f1, f2` et.c., at the top, where f1 and f2 are functions in m.py that are to be exported.
E. Create the file m.py and do nothing: all Python files can be used as modules.

## Part B: Coding questions

*Please use a separate piece of paper (or several) for each question in part B.*

**11**. You can sometimes see statements like

```
assert x > 0
```

in Python code. What does it mean? (1p)

**12**. The code below describes a function that prints a dictionary containing pairs of strings. The output is supposed to give a straight left margin to both columns. The example run shows the intended output. Unfortunately, there are at least two errors in the code. What errors? (2p)

```
def tabulate_dict(d):
    '''
    Tabulate the dictionary d, assumed to contain strings as keys and values.
    The right column is properly left indented.
    '''
    for key, val in d.items():
        if len(key) > left_len:
            left_len = len(key)

    for key, val in d.items():
        n_spaces = len(key)
        spacer = ' ' * n_spaces
        print(key, spacer, val)
```

**Example run, assuming correct code:**

```
[In: ] butterflies = {
    'Sorgmantel': 'Nymphalis antiopa',
    'Nässelfjäril': 'Aglais urticae',
    'Grönsnabbvinge': 'Callophrys rubi',
    'Amiral': 'Vanessa atalanta',
    'Citronfjäril': 'Gonepteryx rhamni',
    }
[In: ] tabulate_dict(butterflies)
[Out:]
Sorgmantel      Nymphalis antiopa
Nässelfjäril    Aglais urticae
Grönsnabbvinge  Callophrys rubi
Amiral          Vanessa atalanta
Citronfjäril    Gonepteryx rhamni
```

**13**. Write the function `count_swedish_alphabet(s)` that counts and returns the number of letters from the Swedish alphabet in s. (1p)

**Exampel run:**

```
[In: ] count_swedish_alphabet('')
[Out:] 0
[In: ] count_swedish_alphabet('åäö')
[Out:] 3
[In: ] count_swedish_alphabet('abc')
[Out:] 3
[In: ] count_swedish_alphabet('AbCdefghijklmnopqrstuvwxyzÅÄÖ)
[Out:] 29
[In: ] count_swedish_alphabet('_(}01:')
[Out:] 0
```

**14**. In this assignment you should a function that counts empty lines. You may give code that solves both subassignments at the same time.

A. Write the function `count_empty_lines(filename)` that returns the number of empty lines in the file given by `filename`.

An empty line is defined as containing only the newline character. If the file does not have any contents, we say that it contains zero lines. If it contains a single newline character, then it contains one empty line. (1p)

B. Extend the function `count_empty_lines(filename)` to handle errors. If the file does not exist or cannot be read, then a warning should be printed ("Warning: could not read the file") and 0 should be returned. (1p)

**15**. Write the function `compose(f, g)` that returns a function that computes `f(g(x))`. You should assume that both `f` and `g` take exactly one argument. (1p)

**Example run:**

```
[In: ] def double(x):
           return 2*x
[In: ] def triple(x):
           return 3*x
[In: ] h = compose(double, triple)
[In: ] h(1)
[Out:] 6
[In: ] h(0)
[Out:] 0
[In: ] h(2)
[Out:] 12
```

**16**. Write the class `Polynomial` to represent polynomials. You wrote code in lab 2 to work with polynomials represented by lists, but now a class will provide a better abstraction.

Your class should define `multiply` to take a numerical factor as argument, so that the polynomial coefficients can be scaled by that factor. You should also have `__str__` defined so that printing polynomials work nicely. See the example run for how it should work.

You may assume that the function `poly_to_string(p_list)` from lab 2 is defined for converting a list of coefficients to a string describing the polynomial. That is, you do not need to list the code, but you can simply call it as if it is part of your code. (3p)

**Example run:**

```
[In: ] p1 = Polynomial([3, 2, 1])
[In: ] print(p1)
[Out:] 3 + 2x + x^2
[In: ] p2 = Polynomial([1, 1, 1])
[In: ] print(p2)
[Out:] 1 + x + x^2
[In: ] p2.multiply(7)
[In: ] print(p2)
[Out:] 7 + 7x + 7x^2
```