

OMTENTAMEN
För kursen
Introduktion till Programmering
TIG003

| | |
|-----------------------|-----------------|
| DATUM: | 2021-03-05 |
| TID: | 09:00-13:00 |
| PLATS: | ONLINE |
| Kursansvar: | Alan B. Carlson |
| Förfrågningar: | 031-786 2786 |
| Max poäng: | 60 |
| Betygsskala: | G 36 p, VG 48 p |

| | |
|------------------|------------------|
| Rättning: | Fråga nr: |
| Alan | 1-5 |
| Ingela | 6-8 |
| Andreas | 9-12 |

Inget samarbete är tillåtet. Inlämningen kommer att skickas till URKUND.

1) Satslogik vs Predikatlogik (3p)

Definiera, förklara och exemplifiera följande begrepp:

- a.) värdering
- b.) sanningsvärdestabell
- c.) individområde

2) Mängddefinitioner (2p)

Antag följande definitionsområde:

$$U = \{1, 2, 3, 4, 5\}$$

Visa vilka mängder (vilka element från definitionsområdet) representeras av följande uttryck:

- a. $\exists x (x < y)$
- b. $\forall y (x <= y)$

3) Logisk sanning, logisk falskhet, logisk kontingens (4p)

Visa med sanningsvärdestabeller om följande satser är logiskt sanna, falska eller kontingenta (dvs varken logiskt sann eller logiskt falsk):

- a.) $P \vee Q \leftrightarrow (P \& \neg Q) \vee (\neg P \& Q)$
- b.) $Q \rightarrow P \leftrightarrow P \vee \neg Q$

4) Logisk konsekvens (6p)

Visa med sanningsvärdestabeller om slutsaten (ii) är en logisk konsekvens av premisserna (i):

- a.) (i) $P \rightarrow Q \vee R$ (ii) $P \rightarrow Q$
- b.) (i) $P \rightarrow (Q \rightarrow R), Q, \neg R$ (ii) P
- c.) (i) $\neg P \& \neg Q$ (ii) $\neg(P \& Q)$

5) Predikatlogiks semantik (3p)

Gör om följande påstående till predikatlogiska satser:

- a.) Vissa normmän hatar popcorn med smör
- b.) Inte alla hundar gillar hundmat
- c.) Alla har en far men inte alla är en far

6) Inmatning och typovandling (1+2+3p)

- a.) Ge ett exempel på när man vill konvertera till typen string.
- b.) Vad är det som blir märkligt i koden och varför? Hur går det att lösa problemet?

```
tal1 = input('Skriv in ett decimaltal: ')
tal2 = input('Skriv in ett till decimaltal: ')
summa = tal1 + tal2
```

c. Hur kan man typomvandla till ett heltal och varför händer "vad som händer" när vi försöker omvandla:

- i.) **6.9**
- ii.) **'4.5'**
- iii.) **'g'**

7) Loopar (2+2+2p)

a.) Förklara koden:

```
x = 0
for _ in range(2, 8):
    x += 2
```

b.) Skriv en kod på tre rader som ger följande output:

```
0 0
0 1
0 2
1 0
1 1
1 2
2 0
2 1
2 2
3 0
3 1
3 2
4 0
4 1
4 2
```

c.) Importera `random` och generera en siffra som är vårt rätta svar med `random.randint(1,10)`. Skapa sen en loop som låter en person skriva in ett nytt tal tills de gissar rätt.

8) Villkorssatser (2+2+2p)

a.) Bygg vidare förra uppgiften med en villkorssats som berättar tips på om siffran personen skrivit in är större eller mindre än den rätta siffran. (Om du inte har gjort förra uppgiften: anta att rätt svar är 3 och ta bara in en gissning per gång du kör programmet)

b.) Anta att bool-variablerna **red** och **blue** kan ha värdet **True** eller **False**. I vilka fall skrivs de olika meningarna ut? Hur kan vi flytta om koden för att den ska fungera mer som tänkt.

```
if blue:
    print('blått ljus')
elif red:
    print('rött ljus')
elif blue & red:
    print('magenta ljus')
else:
    print('inget ljus')
```

c.) Skriv om koden ovan på ett vis att den inte använder någon elif, men fortfarande inte kan skriva ut mer än ett av fallen

9) Datastruktur: Tupler (6p)

Antag att du har en tabell över befolkningsstorlekar i olika städer, enligt:

| | |
|------------|---------|
| Birmingham | 1145000 |
| Leeds | 789000 |
| Sheffield | 580000 |
| Manchester | 553000 |

Utgå från att du får denna tabelldata lagrad i två listor:

```
cities = ["Birmingham", "Leeds", "Sheffield", "Manchester"]  
populations = [1145000, 789000, 580000, 553000]
```

Du vill nu lagra varje tabellrad som en tupel och sedan samla alla rader (tupler) i en lista. Dvs. enligt formen:

```
[ (Birmingham, 1145000) , (Leeds, 789000), ... ]
```

Skriv kod som åstadkommer detta.

10) Funktioner (3+3p)

a) Om vi upprepade gånger behöver fråga en användare om namn och mailadress så kan det vara bra att lägga ut detta i en funktion som kan återanvändas. Skriv en funktion som kan användas till detta och returnera informationen den har fått in.

b) Skriv en funktion som kan användas som en enkel valutakonverterare. Funktionen skall ta in ett värde i Euro och returnera motsvarigheten i SEK. Använd nuvarande växelkurs där 1 Euro = 10,13 SEK (kom ihåg att Python använder punkt istället för decimalkomma).

11) Listor (6p)

Skriv kod som frågar efter längden på 10 personer och lagrar denna data i en lista. Räkna sedan ut medellängden genom att addera alla värden och dela totalsumman på antalet element i listan.

12) Dictionaries (2 + 4p)

a.) Skriv kod som frågar användaren efter användarnamn och lösenord och lagrar detta i ett dictionary med strukturen: { "user": "password"}.

b.) Be nu användaren "logga in" genom att skriva in användarnamn och lösenord. Om uppgifterna matchar det som finns lagrat i dictionary skriver du ut "välkommen", i annat fall "access denied".