Norges teknisk–naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

TDT4110 IT Grunnkurs
Høst 2014

Audietorieøving 1

1 Teori

Oppgave	a	b	c	d	e
Svar	c	b	b	b	c

2 Variabler og verdier

a) variabel_navn = "verdi" Dette kalles en tilordningsetning. Variablnavnet til venstre for likhetstegnet og verdien til høyre. Man trenger ikke spesifisere datatype i python.

b) Når denne koden kjøres vil følgende printes: 3

c)
1. Telefonnummer kan være nyttig både som **heltall** og **streng** ettersom man sjelden trenger å gjøre matematiske operasjoner på et telefonnummer. Begge disse godkjennes.
2. Nedbør i mm vil oftest kunne være **heltall**, men en **desimaltall** kan i enkelte tilfeller være bedre hvis man må kunne operere med større nøyaktighet. Begge disse godkjennes.
3. Hvorvidt en deltaker er påmeldt eller ikke bør lagres som en **sannhetsverdi** (boolsk data)
4. **Streng** er nok best for å lagre en tekstmelding.
5. Prisen på bensin bør lagres som **desimaltall**.

3 Funksjoner og IO

a) Her trenger studenten ikke bruke input til noe for å få godkjent oppgaven.

Kodesnutt 1

```
x = input("Melding til brukeren")
```

Kodesnutt 2

b)
```
streng = "42"
int(streng)
```

4 Betingelser og bruk av logiske uttrykk

Kodesnutt 3

a)
```
if a > b:
  print(a)
```

Kodesnutt 4

b)
```
if a % 3 == 0:
  print('foo')
```

5 Løkker

Kodesnutt 5

a)
```
for i in range(1, 3001):
  betal_middag(i)
```

b) Antar utifra oppgaveteksten at kunde nummer 2400 ønsker å kjøpe middag, men at samtlige påfølgende kunder ikke ønsker å kjøpe middag.

Kodesnutt 6

```
for i in range(1, 3001):
  betal_middag(i)
  if i == 2400:
    break
```

c) Denne kan løses på flere måter, og her har vi listet opp to mulige løsninger:

Kodesnutt 7

```
for i in range(1, 3001):
  if i < 2400:
    betal_middag(i)
  else:
    if(i % 2 == 0):
      betal_middag(i)
    else:
      betal_salat(i)
```

Kodesnutt 8

```
for i in range(1, 3001):
  if i < 2400 or i % 2 == 0:
    betal_middag(i)
  else:
    betal_salat(i)
```

6 Hjernetrim

a) Om vi lister opp alle positive tall under 10 som er delelig med 3 eller 5 får vi 3, 5, 6 og 9. Summen av disse tallene er 23. Skriv kode som gjør det samme for et hvilket som helst heltall n.

Kodesnutt 9: Løsning

```
def morsomfunksjon(n):
  result = 0
  for i in range(n):
    if i % 3 == 0 or i % 5 == 0:
      result += i
  return result

print(morsomfunksjon(10))
```

b) *Collatz conjecture* sier følgende:

- Ta et hvilket som helst naturlig tall (n).
- Om det er et partall: Del det på to ($n/2$).
- Om det er et oddetall: Gang det med 3 og legg til 1 ($3n + 1$).

Om dette gjøres tilstrekkelig mange ganger, vil du alltid ende opp på 1.
Skriv kode som tester denne konjekturen for et hvilket som helst heltall, og stopper kjøringen om $n = 1$.

Kodesnutt 10: Løsning

```
def collatz(n):
  while abs(n) - 1.0 > 0.0001:
    if n % 2 == 0:
      n //= 2
    else:
      n = 3 * n + 1
  return n

print(collatz(4345))
```

c) Fibonaccitallene er definert som følger.

$$f_n = \begin{cases} f_{n-1} + f_{n-2} & \text{hvis } n > 1 \\ 1 & \text{hvis } n = 1 \\ 0 & \text{hvis } n = 0 \end{cases}$$

Skriv kode som summerer opp alle partall i Fibonaccirekken, der siste verdi i rekken ikke skal overstige 20 siffer.

Kodesnutt 11: Løsning

```
def fib():
  a = 0
  b = 1
  fib_sum = 0
  while b < 10 ** 19:
    temp = a
    a = b
    b = temp + b

    if b % 2 == 0:
      fib_sum += b

  return fib_sum

print(fib())
```