



# EKSAMENSOPPGAVE

Institutt: Institutt for matematiske realfag og teknologi

Eksamen i: INF120 Programmering og databehandling  
*emnekode* *emnenavn*

Tid: 15 mai 09:00- 12:00  
*ukedag og dato* *kl. fra – til og antall timer*

Faglærer: Ingunn Burud, 64945705 / 40219286  
*Navn, tlf*

## Tillatte hjelpemidler:

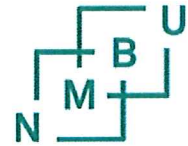
**A1: ingen kalkulator, ingen andre hjelpemidler**

Opgaveteksten er på: 6  
*antall sider inkl. vedlegg*

**Består eksamensoppgaven av deloppgaver, skal det opplyses om hvor mye hver av disse teller.**

Emneansvarlig: Hans E. Plessner 

Sensor: Joakim Sundnes



### Spørsmål 1: Hva skrives ut med følgende kommandoer ? 10 poeng

- a) `print 's = %.2f degrees' % 5.66667`
- b) `print 'hello\nworld'`
- c) `print int(5.667)`
- d) `print float('hello')`
- e) `print eval('3+5')`

### Spørsmål 2: Boolsk uttrykk, 10 poeng

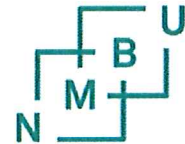
Skriv True eller False når det er gitt at  $x = -5.0$ ;  $y = 4.0$

A)	$x \leq 2$ and $y < -3$
B)	$x < 0$ or $y < 0$
C)	$x < 5$ and $y < 1$
D)	not ( $x < 2$ or $y < -2$ )
E)	$x > -6.0$ and $y > -4$ .

### Spørsmål 3 : If-test 5 poeng

Programmer formelen nedenfor i en Python-funksjon. Gjør koden så effektiv som mulig. Full poengsum gis for den mest effektive løsningen. En mindre effektiv løsning gir 3 poeng.

$$F(x) = \begin{cases} 3 - x, & \text{for } x \leq -5 \\ 2 - x, & \text{for } x < 1 \\ 10, & \text{for } x = 1 \\ x + 5, & \text{for } x \leq 10 \\ x - 5, & \text{for } x > 10 \end{cases}$$



#### Spørsmål 4 : Dictionary 10 poeng

Datafilen densities.txt ser slik ut :

```
air      0.0012
gasoline 0.67
ice      0.9
water    1.0
seawater 1.025
limestone 2.6
granite  2.7
iron     7.8
silver   10.5
mercury  13.6
gold     18.9
platinum 21.4
```

Skriv et program som leser inn denne filen og returnerer en dictionary med de forskjellige materialene som 'keys' og tetthetsverdien som value til hver av de.

Gjør koden din mest mulig effektiv, kort og lesbar. Maksimum poengsum gis for effektiv og lesbar kode.

#### Spørsmål 5: Lister, 5 poeng

Vi har listene:

```
lst = [10, 20, 30, 40, 50]
lst2 = [5, 25, 10, 50, 65]
```

Skriv et program som går gjennom de to listene, og for hvert element sammenligner verdien fra lst1 og lst2 og legger maksimalverdien til en ny liste lst3.

For eksempel er maksimalverdien av 10 og 5 lik 10, av 20 og 25 lik 25 osv. Listen lst3 vil da se slik ut :

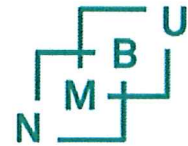
```
[10, 25, 30, 50, 65]
```

#### Spørsmål 6: Numpy arrays, 15 poeng

(I denne oppgaven er numpy importert som np)

- a) Skriv en kommando med numpy i Python som lager en array a, som gir følgende resultat :

```
array([ 11., 12., 13., 14., 15., 16., 17., 18., 19., 20., 21.,
       22., 23., 24., 25., 26., 27., 28., 29., 30.])
```



- b) Gitt array a i oppgaven over, hvordan vil array b se ut ?  
`b = a.reshape(5,4)`

Vi har matrisen:

```
A = [0, 1, 2, 3]
     [4, 5, 6, 7]
     [8, 9, 10, 11]
     [12, 13, 14, 15]
     [16, 17, 18, 19]
```

- c) Hva skrives ut ?

```
print A[:, :-2]
```

- d) Hva skrives ut ?

```
print A[:, :2, 1:3]
```

- e) Skriv en kommando som erstatter alle negative tall i array c nedenfor med tallet 10.

```
c = np.array([ 11., 12., -4., 14., 15., 16., -14., -6., 19.,
              20.])
```

## Spørsmål 7 : Kaste terning, 15 poeng

a) Skriv et program som simulerer et kast med to terninger (vanlige terninger med 1-6 øyne) og regner ut summen av antall øyne. Deretter skal programmet spørre en bruker om å gjette hva summen ble. Dersom brukeren gjetter for lavt skal programmet skrive ut 'Gå høyere', hvis det var for høyt skal det skrive 'Gå lavere'. Dersom tallet er riktig skal programmet skrive 'Riktig' og deretter skrive ut hvor mange forsøk brukeren brukte for å finne riktig svar

Husk at `random.randint(n,m)` henter et integer-slumptall fra og med n, til og med m. Ta med nødvendige import.

b) Utvid programmet til å sjekke med en `try – except` –kommando om tallet som tastes inn fra brukeren er et heltall. Dersom det ikke er et heltall skrives en melding om at kun heltall kan skrives inn.

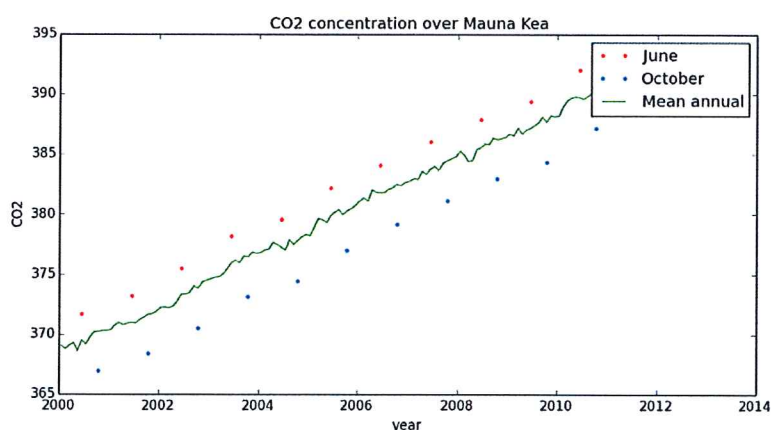


## Spørsmål 8 : Les inn fil og plot data 10 poeng

I denne oppgaven skal du plotte data over CO<sub>2</sub>-fluks fra Mauna Kea. Datafilen `co2.txt` ser slik ut

```
# decimal average interpolated trend #days
# date (season corr)
2000 1 2000.042 369.25 369.25 369.06 27
2000 2 2000.125 369.50 369.50 368.79 23
2000 3 2000.208 370.56 370.56 369.04 31
2000 4 2000.292 371.82 371.82 369.29 29
2000 5 2000.375 371.51 371.51 368.64 29
2000 6 2000.458 371.71 371.71 369.49 29
2000 7 2000.542 369.85 369.85 369.18 28
2000 8 2000.625 368.20 368.20 369.78 28
2000 9 2000.708 366.91 366.91 370.19 30
2000 10 2000.792 366.99 366.99 370.22 31
2000 11 2000.875 368.33 368.33 370.29 25
2000 12 2000.958 369.67 369.67 370.31 30
2001 1 2001.042 370.52 370.52 370.36 29
2001 2 2001.125 371.49 371.49 370.77 28
2001 3 2001.208 372.53 372.53 370.97 29
2001 4 2001.292 373.37 373.37 370.79 30
2001 5 2001.375 373.82 373.82 370.92 25
.....OSV.
```

Du skal plotte kolonne 3 om angir datoen (i desimaltall) mot kolonne5 (interpolated). Du skal ikke plotte hele tidsserien men bare punktene for juni og oktober (angitt som måned 6 og 10). Det vil si at alle data på rad 6 og hver tolvte rad etter det skal plottes. Tilsvarende for rad 10 og og hver tolvte rad etter. I samme vindu skal du plotte hele kolonne 3 mot hele kolonne 6 (trend). Sett også gjerne på såkalte 'legends' i figuren. Figuren skal se slik ut:





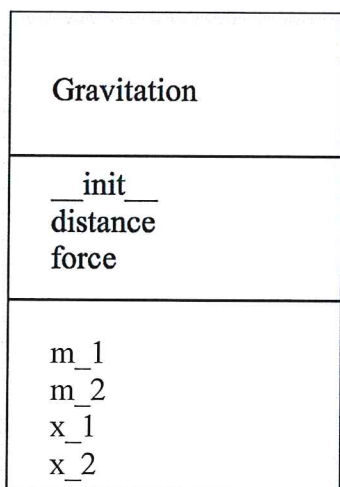
### Spørsmål 9: Klasser 5 poeng

Du skal skrive en klasse som representerer et to-legemers system i en dimensjon. Til konstruktøren (constructor) skal oppgis massen og posisjonen til de to objektene som argumenter ( $m_1$ ,  $x_1$ ,  $m_2$ , og  $x_2$ ). Klassen skal ha en metode som beregner avstanden (def `distance()`), og en som gir gravitasjonskraften mellom dem (def `force()`). beregner gravitasjonskraften mellom to objekter med en gitt masse hver. Formelen for å regne ut denne gravitasjonskraften er :  $F = \frac{GMm}{r^2}$ , der

$$G = 6.67428E11 \text{ m}^3/\text{kg}/\text{s}^2$$

der  $r$  er avstanden mellom de to objektene.

UML til klassen ser slik ut :



### Spørsmål 10 : Rekursjon 10 poeng

Følgende Python-funksjon regner ut summen av en liste slik at kallet `listsum([1, 2, 3, 4, 5])` gir summen 15.

```
def listsum(numlist):  
    sum = 0  
    for i in numlist:  
        sum = sum + i  
    return sum
```

Skriv om denne til en funksjon som bruker rekursjon for å regne ut summen

```
def listsum2(numlist):
```