

1.1 Oppgave 1a

Hva er verdien til variabelen tall etter at følgende kode er utført?

```
tall = 1+2+3  
tall = tall*2 - 1
```

Maks poeng: 1

1.2 Oppgave 1b

Hva er verdien til variabelen tekst etter at følgende kode er utført?

```
tall = 4*2  
t1 = "a"  
if tall>9:  
    t1 = t1 + "a"  
tekst = "b" + t1
```

Maks poeng: 1

1.3 Oppgave 1c

Hva skrives ut til terminalen når følgende kode kjøres?

```
if (5>7) or (2>1):  
    print("ja")  
else:  
    print("nei")
```

Maks poeng: 1

1.4 Oppgave 1d

Hva skrives ut til terminalen når følgende kode kjøres?

```
liste = [1,2,-3,4,-5,6]
total=0
i = 0
while liste[i]>0:
    total += liste[i]
    i += 1
i=len(liste)-1
while liste[i]>0:
    total += liste[i]
    i -= 1
print(total)
```

Maks poeng: 3

1.5 Oppgave 1e

Hva skrives ut til terminalen når følgende kode kjøres?

```
a=3
while a<25:
    a = a*2
    for b in [1,2,3]:
        a += b
print(a)
```

Maks poeng: 3

1.6 Oppgave 1f

Hva skrives ut til terminalen når følgende kode kjøres?

```
tall = 0
operasjoner = "idd.d..did"
for operasjon in operasjoner:
    if operasjon == "d":
        tall = tall*2
    elif operasjon == "i":
        tall = tall+1
    else:
        tall = 0
print(tall)
```

Maks poeng: 3

1.7 Oppgave 1g

Hva skrives ut til terminalen når følgende kode kjøres?

```
bok = {"a":[3,4,5], "b":[6,7]}  
bok["a"].append(8)  
print( len(bok["a"]) )
```

Maks poeng: 2

2.1 Oppgave 2a

Hva skrives ut til terminalen når følgende kode kjøres?

```
def funk(a, b):  
    return a*b
```

```
c = funk(2+3, 2) * 4  
print(c)
```

Maks poeng: 2

2.2 Oppgave 2b

Hva skrives ut til terminalen når følgende kode kjøres?

```
def funk2(a):  
    return a*3
```

```
b = 2  
c = funk2( funk2(b+1) )  
print(c)
```

Maks poeng: 2

2.3 Oppgave 2c

Hva skrives ut til terminalen når følgende kode kjøres?

```
class A:
    def __init__(self, a, b):
        self._c = b
        self._d = a
    def sett_c(self, ny):
        self._c = ny
    def hent_sum(self):
        return self._c + self._d

a1 = A(1,2)
a2 = A(3,4)
a1.sett_c(5)
print( a1.hent_sum() * a2.hent_sum() )
```

Maks poeng: 3

2.4 Oppgave 2d

Hva skrives ut til terminalen når følgende kode kjøres?

```
class A:
    def __init__(self, verdi):
        self._verdi = verdi
    def hent_okt(self):
        return self._verdi+1

class B:
    def __init__(self, a):
        self._a = a
    def hent_modifisert(self):
        return self._a.hent_okt() * 2

def hovedprogram():
    a1 = A(5)
    b1 = B(a1)
    print(b1.hent_modifisert())
```

hovedprogram()

Maks poeng: 3

3.1 Oppgave 3a

Knut vil beregne pris inkludert frakt når han skal handle på nettet.

Skriv en funksjon **pris_inkl_frakt(varepris)**.

Dersom varepris er over 1000 kroner legges det ikke på noe kostnad til frakt og funksjonen skal returnere samme varepris som den fikk inn som argument. Dersom varepris er mellom 500 og 1000 kroner (fra og med 500, til og med 1000) skal det legges på frakt, slik at funksjonen returnerer en pris som er 50 kroner høyere enn varepris. Dersom varepris er under 500 kroner skal det både legges på frakt og ekstragebyr, slik at funksjonen skal returnere en pris som er 80 kroner høyere enn varepris.

Med andre ord skal f.eks. kallet `pris_inkl_frakt(300)` evaluere til 380, `pris_inkl_frakt(600)` evaluere til 650, og `pris_inkl_frakt(1300)` evaluere til 1300.

Skriv ditt svar her...

Maks poeng: 5

3.2 Oppgave 3b

Per vil lage seg en handleliste og raskt kunne fjerne utsolgte varer fra denne lista.

Skriv en funksjon **fjern_utsolgte(handleliste, utsolgte)**.

Funksjonen tar som argumenter en liste med navn på varer (`handleliste`: liste av streng-verdier) og en liste over utsolgte varer (`utsolgte`: liste av streng-verdier). Funksjonen skal returnere en ny liste som inneholde de ønskede varene som ikke er utsolgte, dvs en ny liste som inneholder alle streng-verdier som finnes i listen `handleliste`, men ikke finnes i listen `utsolgte`. I tillegg skal funksjonen skrive til terminalen navnet på hver vare fra handlelisten som er utsolgt.

Altså skal for eksempel kallet `fjern_utsolgte(["melk", "brus", "pasta"], ["kanel", "brus"])` evaluere til listen `["melk", "pasta"]`, samtidig som teksten `brus` skrives til terminalen.

Skriv ditt svar her...

Maks poeng: 5

3.3 Oppgave 3c

Lise skal reise som backpacker gjennom flere land i asia og vil finne ut hvilke vaksiner hun må ta før reisen.

Skriv en funksjon **samlet_vaksinasjon(krav_hvert_land)**.

Funksjonen tar som argument en liste av lister hvor hver indre liste er hvilke vaksiner som trengs for et av landene hun skal reise til. Funksjonen skal returnere en samlet liste over hvilke vaksiner som trengs, dvs en liste av streng-verdier som inkluderer alle vaksinene funnet i et av landene. Dersom samme vaksine finnes i lista for mange ulike land, skal den bare være med én gang i den returnerte lista.

Altså skal for eksempel kallet `samlet_vaksinasjon([["difteri", "tyfoid"], ["hepatit", "difteri"]])` evaluere til en liste `["difteri", "tyfoid", "hepatit"]` (rekkefølgen av vaksinene i den returnerte lista er uviktig).

Skriv ditt svar her...

Maks poeng: 5

3.4 Oppgave 3d

Koden under definerer en variabel setning og lager varianter av denne setningen hvor først ordet "en" er fjernet og deretter også ordet "skal" er fjernet. Det er store likheter mellom kodelinjene som lager setning uten ordet "en" og kodelinjene som lager setning uten ordet "skal". Skriv en alternativ versjon av programmet som unngår denne redundansen ved å definere en funksjon forkort_setning og som kaller denne funksjonen to ganger for å lage setning uten henholdsvis ordene "en" og "skal". Skriv det fulle resulterende programmet, inkludert definisjon av funksjonen (og dens parametre), kall på funksjonen (med argumenter) og andre kodelinjer som trengs for å oppnå samme funksjonalitet som den opprinnelige koden under.

```
setning = "en krabbe skal en dag ut av skallet "
```

```
#fjerner alle ord "en":
setning_v2 = ""
for ord in setning.split():
    if not ord=="en":
        setning_v2 = setning_v2 + ord + " "
```

```
#fjerner alle ord "skal":
setning_v3 = ""
for ord in setning_v2.split():
    if not ord=="skal":
        setning_v3 = setning_v3 + ord + " "
```

```
print(setning_v3) #krabbe dag ut av skallet
```

Skriv ditt svar her...

Maks poeng: 5

4.1 Oppgave 4a)

Denne oppgaven består av en rekke deloppgaver der du skal implementere komponenter av et større system. Om du hopper over en deloppgave er det likevel viktig at du leser hele denne teksten (den ligger samlet som en pdf-fil vedlagt hver deloppgave).

Skriv svar på deloppgave a) fra vedlegget her:

Maks poeng: 5

4.2 Oppgave 4b)

Denne oppgaven består av en rekke deloppgaver der du skal implementere komponenter av et større system. Om du hopper over en deloppgave er det likevel viktig at du leser hele denne teksten (den ligger samlet som en pdf-fil vedlagt hver deloppgave).

Skriv svar på deloppgave b) fra vedlegget her:

Maks poeng: 5

5.1 Oppgave 5)

Karl vil kunne sikre seg at han er forskånt for å måtte lese et bestemt fy-ord og dets synoner.

Skriv en funksjon `sjekk_om_fyord(setning, fyord, synonym_liste)`.

Funksjonen tar som argument en setning (en streng bestående av flere ord med mellomrom mellom) , et fyord (en streng bestående av ett enkelt ord uten noen mellomrom) og en samling av ulike synonymer (en liste av lister, hvor hver indre liste består av streng-verdier som er enkeltord). Funksjonen skal sjekke om minst ett av ordene i setningen er et synonym med det oppgitte fyordet (ifølge den oppgitte samlingen av synonymer) eller om fyordet i seg selv finnes i setningen. Dersom fyordet eller et synonym av fyordet finnes i setningen skal funksjonen returnere True. Hvis ikke skal funksjonen returnere False.

Altså skal for eksempel kallet `sjekk_om_fyord("spis masse godsaker", "snop", [{"saft", "lemonade"}, {"snacks", "snop", "godsaker"}, {"mye", "masse"}])` returnere True, mens kallet `sjekk_om_fyord("spis masse godsaker", "lemonade", [{"saft", "lemonade"}, {"snacks", "snop", "godsaker"}, {"mye", "masse"}])` skal returnere False.

Skriv ditt svar her...

Maks poeng: 6