

1 1a

```
class Klasse:  
    def __init__(self, tall):  
        self._tall = tall  
  
x = Klasse(3)  
y = Klasse(4)  
z = Klasse(5)  
u = x  
v = u  
u = None
```

Hvor mange Klasse-objekter refereres fra variabler i koden på bildet (etter at alle linjene er blitt utført)? (3).

Maks poeng: 1

2 1b

Hva blir resultatet av å kjøre denne koden?

- IndexError: list index out of range
- 6 skrives ut
- 4 skrives ut
- 5 skrives ut



```
a=1
b=2
c=3
listeA = [a,b,c]
listeB = [1,2,3]
listeC = listeA
listeA.append(5)
listeC.append(6)
print(listeC[4])
```

Maks poeng: 1

3 1c

```
# Ordbok med statene i USA:
ordbok = {
    "AL": "Alabama", "AK": "Alaska", "AZ": "Arizona", "AR": "Arkansas", "CA": "California", \
    "CO": "Colorado", "CT": "Connecticut", "DE": "Delaware", "FL": "Florida", \
    "GA": "Georgia", "HI": "Hawaii", "ID": "Idaho", "IL": "Illinois", "IN": "Indiana", \
    "IA": "Iowa", "KS": "Kansas", "KY": "Kentucky", "LA": "Louisiana", \
    "ME": "Maine", "MD": "Maryland", "MA": "Massachusetts", "MI": "Michigan", \
    "MN": "Minnesota", "MS": "Mississippi", "MO": "Missouri", "MT": "Montana", \
    "NE": "Nebraska", "NV": "Nevada", "NH": "New Hampshire", "NJ": "New Jersey", \
    "NM": "New Mexico", "NY": "New York", "NC": "North Carolina", "ND": "North Dakota", \
    "OH": "Ohio", "OK": "Oklahoma", "OR": "Oregon", "PA": "Pennsylvania", \
    "RI": "Rhode Island", "SC": "South Carolina", "SD": "South Dakota", "TN": "Tennessee", \
    "TX": "Texas", "UT": "Utah", "VT": "Vermont", "VA": "Virginia", "WA": "Washington", \
    "WV": "West Virginia", "WI": "Wisconsin", "WY": "Wyoming", "AS": "American Samoa", \
    "DC": "District of Columbia", "FM": "Federated States of Micronesia", "GU": "Guam", \
    "MH": "Marshall Islands", "MP": "Northern Mariana Islands", "PW": "Palau", \
    "PR": "Puerto Rico", "VI": "Virgin Islands" }

b= []
for a in ordbok:
    b.append(a)

for i in range(len(ordbok)):
    a = b[i]
    ordbok[a] = ordbok[b[i]]

print(ordbok[b[1]])
```

Hva skrives ut på terminalen av koden i bildet? Fyll ut svarboksen med nøyaktig samme tegn:

 (Alaska)

Maks poeng: 1

4 1d

```
# Ordbok med statene i USA:
ordbok = {
    "AL": "Alabama", "AK": "Alaska", "AZ": "Arizona", "AR": "Arkansas", "CA": "California", \
    "CO": "Colorado", "CT": "Connecticut", "DE": "Delaware", "FL": "Florida", \
    "GA": "Georgia", "HI": "Hawaii", "ID": "Idaho", "IL": "Illinois", "IN": "Indiana", \
    "IA": "Iowa", "KS": "Kansas", "KY": "Kentucky", "LA": "Louisiana", \
    "ME": "Maine", "MD": "Maryland", "MA": "Massachusetts", "MI": "Michigan", \
    "MN": "Minnesota", "MS": "Mississippi", "MO": "Missouri", "MT": "Montana", \
    "NE": "Nebraska", "NV": "Nevada", "NH": "New Hampshire", "NJ": "New Jersey", \
    "NM": "New Mexico", "NY": "New York", "NC": "North Carolina", "ND": "North Dakota", \
    "OH": "Ohio", "OK": "Oklahoma", "OR": "Oregon", "PA": "Pennsylvania", \
    "RI": "Rhode Island", "SC": "South Carolina", "SD": "South Dakota", "TN": "Tennessee", \
    "TX": "Texas", "UT": "Utah", "VT": "Vermont", "VA": "Virginia", "WA": "Washington", \
    "WV": "West Virginia", "WI": "Wisconsin", "WY": "Wyoming", "AS": "American Samoa", \
    "DC": "District of Columbia", "FM": "Federated States of Micronesia", "GU": "Guam", \
    "MH": "Marshall Islands", "MP": "Northern Mariana Islands", "PW": "Palau", \
    "PR": "Puerto Rico", "VI": "Virgin Islands" }
liste = []
for a in ordbok:
    liste.append(a)
i = 0
for b in liste:
    if ordbok[b] == "Alabama":
        print(liste[i])
    i += 1
```

Hva skrives ut på terminalen av denne koden? Svar her (nøyaktig de tegnene som skrives ut):

(AL)

Maks poeng: 1

5 1e

```
s = ""
if True or False:
    s += "a"
    if True and False:
        s += "b"
        s += "c"
    if False or not False:
        s += "d"
a = False
b = not False
if a or b:
    if a and b:
        s += "e"
if not False and not False:
    s += "f"
    if not True and True:
        s += "g"
    if True or not True:
        s += "h"
print(s)
```

Skriv nøyaktig de tegnene som skrives ut i svarfeltet:

(adfh)

Maks poeng: 1

6 1f

```
s = ""
if True or not False:
    s += "a"
    if True and not False:
        s += "b"
    if False or not False:
        s += "c"
        s += "d"
elif True:
    s += "e"
else:
    s += "f"
a = False
b = not True
if a or b:
    if a :
        s += "g"
print(s)
```

Skriv nøyaktig de tegnene som blir skrevet ut i svarfeltet:

(abcd)

Maks poeng: 1

7 1g

```
class Klasse:
    def __init__(self, a, b):
        self._a = b
        self._b = a

    def hent_a(self):
        return self._a

    def hent_b(self):
        return self._b

c = Klasse("a", "b")
print(c.hent_a() + c.hent_b())
```

Skriv nøyaktig de tegnene som skrives ut på terminalen i svarfeltet:

(ba)

Maks poeng: 1

8 1h

```
class Klasse:
    def __init__(self, a):
        self._a = a
        self._b = "a"

    def hent_a(self):
        return self._a

    def hent_b(self):
        return "b"

c = Klasse("a")
print(c.hent_a() + c.hent_b())
```

Skriv nøyaktig de tegnene som skrives ut på terminalen i svarfeltet:

(ab)

Maks poeng: 1

9 1i

Fyll inn verdien i boksen under som gjør at koden under ville skrevet ut **Kanskje** til terminalen (linjen som starter med `y=` kjøres før koden på bildet).

`y=` (13)

```
assert type(y)==int
z=12
x = (y==z)
if x:
    print("Ja")
elif y>11 and y<14:
    print("Kanskje")
else:
    print("Nei")
```

Maks poeng: 1

10 1j

Fyll inn det korteste mulige svaret (i antall tegn) i boksen under som gjør at koden under ville skrevet ut **11** til terminalen.

```
liste = [3,4,5,6]
```

```
a=4
```

```
for x in liste:
```

```
    if x  (<=) 4:
```

```
        a += x
```

```
print(a)
```

Maks poeng: 1

11 2a

```
class Klasse:
    def __init__(self, tall):
        self._tall = tall

    def sett_tall(self, tall):
        self._tall = tall

    def skriv_tall(self):
        print(self._tall)

x = Klasse(3)
y = Klasse(4)
z = x
u = Klasse(5)
u = z
z.sett_tall(2)
u.skriv_tall()
```

Hva skrives ut på terminalen av koden på bildet? (2).

Maks poeng: 2

12 2b

```
def funk(ordbok, ord, tall):  
    ordbok[ord] = tall  
    return 2*tall  
  
bokA = {"a":3, "b":6, "c":8, "d":10}  
bokB = {"a":3, "b":6, "c":8, "d":10}  
c = "b"  
d = c  
x = 2  
y = funk(bokB, c, 3)  
print(bokB[d])
```

Fyll inn ditt svar her:

(3)

Maks poeng: 2

13 2c

Fyll inn et tall i svarboksen som gjør at koden under ville skrevet ut 43 til terminalen (linjen som begynner med x= kjøres før koden på bildet).

x= (13)

```
m=0
k=2
while k<x:
    k = k*2
    m=m+10
print(m+x)
```

Maks poeng: 2

14 2d

Fyll inn det som må stå i boksen under for at variabelen d2 ville fått verdien {'a': 3, 'b': 4, 'c': 2} etter at koden under var kjørt.

```
d1 = {"a": [1, 2, 6], "b": [1, 4, 2, 8], "c": [1, 5]}
```

```
d2 = {}
```

```
for key in d1:
```

```
    value = d1[key]
```

```
    d2[key] =  (len(value), len(d1[key]))
```

Her er det flere mulige svar - alle korrekte svar er akseptert for studenter der dette endrer vurdering til Bestått

Maks poeng: 2

15 **2e**

Fyll inn det som må stå i boksen under for at kode under skal skrive **Positiv og dobbel** til terminalen dersom brukeren skriver inn to positive heltall, hvor det første tallet er nøyaktig dobbelt så stort som det andre tallet, og ellers skriver ut **Nei**.

```
t1 = int(input())
t2 = int(input())
t3 = 0
```

```
if (t1>t3) and (t2>t3) and  (t1/t2==2, t1/t2==2.0, t1//t2==2, t2*2==t1,
```

```
t1/2==t2, t1//2==t1, t1/2==t1, 2==t1/t2, 2.0==t1/t2, 2==t1//t2, t1==t2*2, t2==t1/2, t1==t1//2,
t1==t1/2, t2/t1==0.5, (t1/t2)==2, (t1/t2==2), (t1//t2==2)) :
```

```
    print("Positiv og dobbel")
```

```
else:
```

```
    print("Nei")
```

**Her er det flere mulige svar - alle
korrekte svar er akseptert for
studenter der dette endrer vurdering
til Bestått**

Maks poeng: 2

16 2f

Gitt klassen Rektangel:

```
class Rektangel:
    def __init__(self, lengde, bredde):
        self._lengde = lengde
        self._bredde = bredde
        self._areal = lengde*bredde

    def sett_lengde(self, lengde):
        self._lengde = lengde

    def sett_bredde(self, bredde):
        self._bredde = bredde

    def hent_areal(self):
        return self._areal

    def oppdater_areal(self):
        self._areal = self._lengde * self._bredde
```

Fyll inn det som må stå i boksen nedenfor for at koden nedenfor vil skrive ut **6 12** på terminalen.r1= (Rektangel(2,3)) her.

```
areal1 = r1.hent_areal()
r1.sett_lengde(4)
r1.oppdater_areal()
areal2 = r1.hent_areal()
print(areal1, areal2)
```

Maks poeng: 2

17 2g

```

class Klasse:
    def __init__(self, tall):
        self._a = tall
        self._b = 5

    def legg_sammen (self):
        c = self._a + self._b
        return c

    def __str__(self):
        return "a: "+str(self._a) + "; b: "+str(self._b)

d = Klasse(3)
e = str(d)

```

Gitt koden i bildet, velg den beskrivelsen som passer best for hver rad.

Velg beste alternativ for hver rad.

	lokal variabel	representasjon av objekt	parameter	instansvariabel
self_a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓
tall	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>
e	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
c	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Maks poeng: 2

18 2h

	Uttrykk som evaluerer til int	Ikke et gyldig uttrykk	Uttrykk som evaluerer til string
713	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
for i in range(4):	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>
int(input("Alder: "))	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
print(navn)	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>

Maks poeng: 2