```
class Gruppetime:
    def __init__(self):
        self.fag = "IN1000“
    def __str__(self):
        return self.fag
g = Gruppetime()
print(g)
```



Hva er en klasse?

```
class Gruppetime:
    ...


g1 = Gruppetime()
g2 = Gruppetime()
```

Konstruktør: `__init__`

- Må ikke ha dette, men gjør det enkelt å opprette objekter av klasser

```
class Gruppetime:
    def __init__(self, fag, rom):
        self.fag = fag
        self.rom = rom

g1 = Gruppetime("IN1000", "CAML")
g2 = Gruppetime("IN2110", "JAVA")
```

Instansvariabler

- Variabler som er unik for hver instans av en klasse

```
class Gruppetime:
    def __init__(self, fag, rom):
        self.fag = fag
        self.rom = rom
```

```
class Gruppetime:
    def __init__(self):
        self.fag = "IN1000“
    def __str__(self):
        return self.fag

g = Gruppetime()
print(g)
```

self som parameter

Hvorfor det?

Printe ut klassen: `__str__`

- Må ikke ha dette, men gjør det enkelt å printe ut klassen

Hvis vi ikke har `__str__` :

```
class Gruppetime:
    def __init__(self, fag, rom):
        self.fag = fag
        self.rom = rom

g1 = Gruppetime(“IN1000”, “CAML”)
g2 = Gruppetime(“IN2110”, “JAVA”)
print(g1)
```

Så printes:

```
<__main__.Gruppetime object at 0x0000022550E69BB0>
```

Hva betyr dette?

Hvis vi har __str__ :

```
class Gruppetime:
    def __init__(self, fag, rom):
        self.fag = fag
        self.rom = rom
    def __str__(self):
        s = “Fag: “ + self.fag + “ Rom: “self.rom”
        return s

g1 = Gruppetime(“IN1000”, “CAML”)
g2 = Gruppetime(“IN2110”, “JAVA”)
print(g1)
```

Så printes:

```
Fag: IN1000 Rom: CAML
```

Klasser kan ha metoder:

```
class Gruppetime:
    def __init__(self, fag, rom):
        self.fag = fag
        self.rom = rom
    def hentFag(self):
        return self.fag
```

```
g1 = Gruppetime(“IN1000”, “CAML”)
g2 = Gruppetime(“IN2110”, “JAVA”)
print(g1.hentFag())
```

Så printes:

```
IN1000
```

#Lag en klasse Student, som har student som parameter i konstruktøren. (hint: __init__)

Student har instansvariablene student og score. Sett score til 0.

#I Student-klassen: Lag metoden leggTilScore, som legger til 1 poeng til score (husk! (self))

#I Student-klassen: Lag en metode som skriver ut all informasjon om klasseobjektet (hint: __str__)

```python
class Student:
    def __init__(self, student):
        self.student = student
        self.score = 0
    def leggTilScore(self):
        self.score += 1
    def __str__(self):
        return "Studenten " + self.student + " har Score: " + str(self.score)
```

Lag et nytt Student-objekt, med deres eget navn

f.eks.

```
s = Student(“Helene”)
#NB! Utenfor Klassen!
```

07.07: Feil i Ekorn-klasse

IN1000 feilmeldinger høst2022 høst2023 høst2024 in1000 klasse konstruktør mengde parameter uke7

I denne oppgaven skal du finne feil i programmet under. Du trenger ikke å tenke på at noen av feilene ikke blir relevante fordi programmet slutter å kjøre tidlig; regn med alle feil du finner.

Prøv først å løse oppgavene uten å kjøre kodesnutten eller kopiere den til en editor!

```
1| class Ekorn:
2|     def _init_(self, pelsfarge, bosted):
3|         self._pelsfarge = pelsfarge
4|         self._bosted = bosted
5|
6|     def hentPelsfarge():
7|         return self._pelsfarge
8|
9|     def hentBosted(self):
10|        return self.bosted
11|
12| ekorn1 = Ekorn("brunt","Drammen")
13| ekorn2 = Ekorn("svart")
14| ekorn3 = ekorn("grått","Tjøme")
```

a)

1. Linje 2: Det er kun én understrek på hver side av init, burde vært `__init__()`.
2. Linje 6: Metoden `hentPelsfarge()` mangler `self` som parameter.
3. Linje 10: Metoden `hentBosted()` henter `self` sin `bosted`, men det må være `_bosted`.
4. Linje 13: `ekorn2` mangler et parameter; parameter `bosted` er obligatorisk.
5. `ekorn3` skriver klassen `ekorn` med liten forbokstav, klasser har alltid stor forbokstav.

```
s.leggTilScore()

print(s)
```

Eksamen 2022

Oppgave 4a) (3 poeng)

Skriv klassen **Gruppe**. Konstruktøren skal ha én parameter; en liste med krav til fasiliteter i rom.
Klassen skal tilby følgende metoder i tillegg til konstruktør:

- **legg_til_personer().** Parameter er en liste med personnavn som legges til gruppen (i tillegg til de som eventuelt var med i gruppen fra før)
- **hent_personer().** Returnerer navnene i gruppen
- **hent_krav()**. Returnerer fasilitetene gruppen krever i sine rom.

Fasit (fra løsningsforslag - eksamen 2022)

```
# 4a) 3 poeng
class Gruppe:
    def __init__(self, krav):
        self._krav = krav                       # Liste av strenger
        self._personer = []                     # Liste av strenger

    def legg_til_personer(self, personer):  # Liste av strenger
        for person in personer:
            self._personer.append(person)

    def hent_personer(self):                    # Returnerer liste av strenger
        return self._personer

    def hent_krav(self):                        # Returnerer liste av strenger
        return self._krav
```

Trix: 07.10 MasterMind-spillet

Eller andre oppgaver - uke 7