Oppgave 1

```
class Rom:
    def __init__(self, kvadratmeter, type):
        self._type = type
        self._størrelse = kvadratmeter

    def type(self):
        return self._type

    def størrelse(self):
        return self._størrelse
```

Oppgaven deres er å lage en klasse for hus bestående av fem rom; en stue, et bad, et kjøkken og to soverom. Størrelsen på rommene velger dere selv. Det skal være mulig å legge til rom, men bare fem og bare romtypene som er nevnt. Dere må finne ut hvilke instansvariabler, metoder og parametere dere trenger selv. Det skal være mulig å kalle `print(et_husobjekt)` og få informasjon om huset og alle rommene i huset skrevet ut til terminalen (inkludert total husstørrelse i kvadratmeter).

Oppgave 2

Klassen `Departement` under innehar data og metoder knyttet til Helsedepartementet (helsedept) og Utdanningsdepartementet (utdanningsdept).

```
class Departement:
    def __init___(self, helse, utdanning):
        self._beskrivelse_helsedept = helse
        self._beskrivelse_utdanningsdept = utdanning
        self._ansatte_helsedept = []
        self._ansatte_utdanningsdept = []

    def legg_til_ansatt_helsedept(self, ny_ansatt):
        self._ansatte_helsedept.append(ny_ansatt)

def legg_til_ansatt_utdanningsdept(self, ny_ansatt):
        self._ansatte_utdanningsdept.append(ny_ansatt)
```

```
    def finn_ansatt_helsedept(self, ansatt_som_søkes):
        for ansatt in self._ansatte_helsedept:
            if ansatt == ansatt_som_søkes:
                return ansatt
        return None

    def finn_ansatt_utdanningsdept(self, ansatt_som_søkes):
       for ansatt in self._ansatte_utdanningsdept:
           if ansatt == ansatt_som_søkes:
               return ansatt
       return None
```

a) Hva er mulige utfordringer ved å strukturere `Departement` på denne måten?

b) Del opp `Departement`-klassen for å imøtekomme utfordringene fra a).

c) Opprett `__str__`-metoder og `__eq__`-metoder for klassene fra b). Her må dere selv velge hva det er naturlig at metodene inneholder.

d) Lag et hovedprogram der dere oppretter objekter av klassene fra b) og tester metodene for å legge til nye og finne ansatte. Test i tillegg at `__str__`- og `__eq__`-metodene fra c) fungerer. Husk at `__str__` og `__eq__` aldri kalles direkte på et objekt.

Oppgave 3

Når innhøstingen av korn og grønnsaker pågår for fullt har bonden ingen traktorer stående på låven. Det vil si at traktorene brukes til arbeid på jordet. Bonden har tre traktorer. Hver traktor har en maksimal trekkvekt (en grense for antall kilo en traktor kan trekke av gangen). Trekkvekten er ulik for hver av de tre traktorene.

Fra september til november trengs det gradvis færre traktorer på jordet samtidig. I løpet av september kjører derfor bonden én av traktorene inn på låven. Etter oktober står det to traktorer på låven, og i løpet av november står alle traktorene der. Traktorene kjøres inn på låven i sortert rekkefølge etter maksimal trekkvekt.

Oppgaven deres er å lage klasser forbundet med bondens innhøsting, basert på informasjonen over. Dere må finne ut hvilke instansvariabler, metoder og parametere

dere trenger selv. Det skal være mulig å hente ut informasjon om traktorene som er på jordet til enhver tid og informasjon om traktorene som står på låven til enhver tid. Vi må kunne finne total trekkvekt som til enhver tid befinner seg på jordet og låven. Lag et hovedprogram som kombinerer objekter av klassene og tilhørende metoder. Hovedprogrammet må oppfylle kravene til funksjonalitet beskrevet i oppgaveteksten.

Oppgave 4

På Stortinget finnes det flere partier: Arbeiderpartiet (Ap), Fremskrittspartiet (FrP), Høyre (H), Kristelig Folkeparti (KrF), Sosialistisk Venstreparti (SV), Venstre (V), Miljøpartiet De Grønne (MDG), Rødt (R). Hvert av disse partiene har flere stortingsrepresentanter.

Et parti kan kun legges til i Stortinget én gang (hvis det ikke finnes der fra før). Hvert parti har et unikt navn. En stortingsrepresentant kan kun legges til i et parti én gang (hvis vedkommende ikke finnes der fra før). Hver stortingsrepresentant har et unikt fødselsnummer (fødselsdato og personnummer).

Oppgaven deres er å lage Stortinget bestående av klasser basert på informasjonen. Dere må finne ut hvilke instansvariabler, metoder og parametere dere trenger selv. Partiobjekter og representantobjekter skal kunne sammenlignes med ==. Et kall på `print(et_partiobjekt)` skal skrive informasjon om alle stortingsrepresentantene knyttet til partiet til terminalen. Kall på `print(stortinget)` skal skrive informasjon om stortingspartiene og deres representanter til terminalen. Lag et hovedprogram som kombinerer objekter av klassene og tilhørende metoder. Hovedprogrammet må oppfylle kravene til funksjonalitet beskrevet i oppgaveteksten.

Oppgave 5

Et sykehus har flere bygninger med forskjellige avdelinger fordelt mellom de ulike bygningene. Hver bygning har en unik bygningsid og hver avdeling har en unik avdelingsid. Hver bygning og hver avdeling har et navn (f.eks. hovedbygningen eller ortopedisk avdeling). Dessuten har hver bygning og hver avdeling en resepsjon, og i

enkelte tilfeller har bygninger og avdelinger samme resepsjon. Hver resepsjon har en resepsjonsid, sammensatt av strengen “resepsjon” og bygningsid og avdelingsid.

Bygningen der all kirurgi finner sted deler resepsjon med avdeling for hjertekirurgi. Dere skal, i tillegg til å bruke `Resepsjon`-klassen under, lage klassene som mangler. Dere må finne ut hvilke instansvariabler, metoder og parametere dere trenger selv. Metoden `_sett_resepsjonsid` (non-public og skal kun brukes i klassen) nedenfor mangler innhold, som dere må skrive. Lag deretter et hovedprogram som sørger for at kirurgibygningen og avdeling for hjertekirurgi får den samme resepsjonen. Dere skal som del av prosessen forsikre dere om at resepsjonen får riktig resepsjonsid.

Basert på hovedprogrammet skal dere også lage en datastrukturtegning som viser dataene i objektenes instansvariabler og sammenhengen mellom objektene. Dere kan blant annet se eksempler på datastrukturtegninger i ukeressurser for uke 8.

```
class Resepsjon:
    def __init__(self, bygning, avdeling):
        self._bygning = bygning
        self._avdeling = avdeling
        self._resepsjonsid = self._sett_resepsjonsid(bygning, avdeling)

    def hent_resepsjonsid(self):
        return self._resepsjonsid

    def hent_bygning(self):
        return self._bygning

    def hent_avdeling(self):
        return self._avdeling

    def _sett_resepsjonsid(bygning, avdeling):
        pass
```