Oppgave 4

I denne oppgaven skal du lage et komplett system som tar for seg emner, studieprogram og deres tilknytning til ulike institutter på universitetet. Et emne er et kurs som studenter kan følge, slik som IN1000. Studieprogram er løp bestående av en rekke emner. Institutter er enheter ved universitetet som eier emner og studieprogram. Ett studieprogram kan inneholde emner fra ulike institutter.

Oppgaven er inndelt i deloppgaver. Hver deloppgave finner du på egen side i Inspera.

Å huske riktig navn på magiske metoder er en del av oppgaven. Men du vil selvsagt få noe uttelling for å kunne lage en magisk metode riktig selv om du bruker feil navn på den.

Du må ikke få til hele deloppgaven for å få poeng. Alle metoder du lager vil telle positivt selv om du ikke har laget alle.

Du kan hoppe over deloppgaver og likevel få til deloppgavene som følger etterpå, ved å bruke informasjon som er gitt i deloppgavene du ikke har gjort.

VIKTIG: Dersom du ikke vet *hvordan* du skal lage en metode, kan du likevel anta at metoden eksisterer og bruke den videre *som om* den virker riktig. Vi bruker for eksempel ofte *print*-funksjonen, uten at vi egentlig vet *hvordan* den får til å skrive til terminalen. Det er i de fleste sammenhenger nok å vite hvordan den *brukes* (litt som at du kan kjøre en bil uten å kunne reparere eller bygge den). Til dette formålet vil opplysningene gitt i tabellene vil være til stor hjelp.

Deloppgave A

Lag en klasse ved navn **Emne** som representerer en enkelt emne. Et **Emne**-objekt skal ha følgende instansvariabler:

Variabelnavn	Type	Kommentar
emnekode	**str**	Emnekoden til emnet (for eksempel «IN1000»
navn	**str**	Navnet til emnet (for eksempel «Introduksjon til objektorientert programmering»)
høst	**bool**	**True** dersom emnet undervises om høsten, **False** hvis ikke
vår	**bool**	**True** dersom emnet undervises om våren, **False** hvis ikke

Videre skal et **Emne**-objekt ha følgende metoder:

Metodenavn	Returverdi	Parametre[1]	Kommentar
(konstruktør)	-	Emnekode (**str**), navn (**str**), om emnet undervises om høsten (**bool**) og om emnet undervises om våren (**bool**)	-
(den magiske metoden som kalles når objektet printes)	**str**	-	Skal returnere informasjon om emnet på formatet som disse eksemplene viser: «IN1000: Introduksjon til objektorientert programmering (høst og vår)» «IN1910: Programmering for naturvitenskapelige anvendelser (høst)»

Deloppgave B

Lag en klasse ved navn **Studieprogram** som representerer en samling av flere **Emne**-objekter. Et **Studieprogram**-objekt skal ha følgende instansvariaber:

Variabelnavn	Type	Kommentar
navn	**str**	Studieprogrammets navn
_data	**dict**	Denne private ordboken skal ha emnekoder (**str**) som nøkler, og **Emne**-objekter som verdier.

Videre skal et **Studieprogram**-objekt ha følgende metoder:

Metodenavn	Returnerer	Parametre	Kommentar
(konstruktør)	-	Navnet til studieprogrammet (**str**)	Lager en tom ordbok til å begynne med.

[1] **self** kommer eventuelt i tillegg til parametrene nevnt her

legg_til	-	**Emne**-objekt	Legger til et nytt **Emne**-objektet i samlingen, hvis emnekoden ikke finnes fra før. Hvis den finnes fra før, skrives det ut en beskjed om at emnekoden allerede eksisterer i samlingen. Om du hoppet over eller ikke fikk oppgave A, kan du likevel anta at denne klassen er riktig laget i denne og resten av deloppgavene.
(den magiske metoden som kalles når man går gjennom **Studieprogram**-objektet med en **for**-løkke)	(Se kommentar)	-	Når en **for**-løkke går gjennom et **Studieprogram**-objekt skal den gå innom alle **Emne**-objektene i samlingen, ett etter ett. (Hint: bruk **yield**, ikke **return.**)
(den magiske metoden som kalles når man vil vite antall elementer i samlingen)	**int**	-	Antall **Emne**-objekter i samlingen.

Deloppgave C

I denne deloppgaven skal du lage et hovedprogram som tester funksjonaliteten i klassene du allerede har skrevet. Hovedprogrammet skal gjøre følgende:

- Lage følgende objekter av typen **Emne**:

    * IN1000 – Introduksjon til objektorientert programmering (høst, vår)

    * IN1010 – Objektorientert programmering (vår)

    * IN1150 – Logiske metoder (vår[2])

    * IN1050 – Introduksjon til design (høst)

[2] Emnet har også eksamen om høsten, men det viktige her er at det kun undervises om våren.

- Lage følgende objekter av typen **Studieprogram**:

    * Programmering og systemarkitektur

    * Design, bruk, interaksjon

- Knytte objektene for IN1000 og IN1010 til begge studieprogrammene over

- Knytte objektet IN1150 til Programmering og systemarkitektur

- Knytte objektet IN1050 til Design, bruk, interaksjon

- Skrive ut hvert studieprogram med navn og antall emner, og skrive ut alle emnene tilknyttet studieprogrammet under hvert studieprogram

Deloppgave D

Lag en klasse ved navn **Institutt** som representerer en samling av flere **Studieprogram**-objekter. Et **Institutt**-objekt skal ha følgende instansvariaber:

Variabelnavn	Type	Kommentar
_studieprogrammer	**list**	Denne private listen skal ha **Studieprogram**-objekter som elementer. Om du hoppet over eller ikke fikk til oppgave B, kan du likevel anta at denne klassen er riktig laget i denne og resten av deloppgavene.
_emner	**dict**	Denne private ordboken skal ha emnekoder (**str**) som nøkler og **Emne**-objekter som verdier. Om du hoppet over eller ikke fikk til oppgave A, kan du likevel anta at denne klassen er riktig laget i denne og resten av deloppgavene.
navn	**str**	Instituttets navn

Videre skal et **Institutt**-objekt ha følgende metoder:

Metodenavn	Returnerer	Parametre	Kommentar
(konstruktør)	-	Navn på instituttet (**str**)	Lager en tom liste og en tom ordbok til å begynne med.
legg_til	-	Et **Studieprogram**-objekt	Legger til **Studieprogram**-objektet i listen og alle dets **Emne**-objekter som verdier i ordboken (med emnekodene som nøkler). Her trenger du ikke sjekke om emnekodene finnes i ordboken fra før.

(den magiske metoden som kalles når man går gjennom **Institutt**-objektet med en **for**-løkke)	(Se kommentar)	-	Når en **for**-løkke går gjennom et **Institutt**-objekt skal den gå innom alle **Studieprogram**-objektene i samlingen.
(den magiske metoden som kalles når man vil vite antall elementer i samlingen)	**int**	-	Antall **Studieprogram**-objekter i samlingen skal telles med én gang, selv om emnekoden finnes i flere studieprogrammer.

Deloppgave E

Utvid klassen Emne (ta gjerne med deg en kopi til denne siden, slik at den opprinnelige blir stående). Klassen skal nå ha en privat instansvariabel **_eier** som kan inneholde en referanse til et objekt av typen **Institutt**. Denne initialiseres til **None**.

Klassen skal ha en metode **legg_til_eier** som setter argumentet til metoden som eier av emnet.

Utvid også den magiske metoden til **Emne** slik at navnet på instituttet som eier emnet er del av strengen som returneres, dersom det finnes en eier. Om eier-variabelen ikke er satt (er lik **None**), skal teksten være den samme som før endringen.

Deloppgave F

Utvid klassen **Institutt** (ta gjerne med deg en kopi til denne siden, slik at den opprinnelige blir stående). Når du kaller **legg_til(studieprogram)** på et objekt av klassen **Institutt**, skal automatisk dette objektet bli eier av alle emnene som tilhører **Studieprogram**-objektet **studieprogram**.

Deloppgave G

Utvid hovedprogrammet ditt (ta gjerne med deg en kopi til denne siden, slik at den opprinnelige blir stående). Du skal nå teste at funksjonaliteten fra oppgave E og F fungerer slik du ønsker.

(Det er lov, men ikke nødvendig, å bruke **assert**.)