Prøveeksamen i IT2 vår 2025

**Lykke til!**

For at vi skal følge retteskjemaet til forrige eksamen, vil det være slik som dette:

**NB! Anbefalt tidsbruk på prøveeksamen er sånn cirka**

* En time og femten minutter på oppgave 1-6 som er 30 % av vektingen.
  + NB! Oppgave 6 er mye mer tidkrevende enn oppgave 1-5.
* En time og femten minutter på oppgave 8 som er 30 % av vektingen
* To timer på oppgave 9 som er 40 % av vektingen
* En halv time til å se over, pakke filer, kommentere, endre på ting (siden vi mangler oppgave 7).

Denne wordfilen med svarene innskrevet, programfilene du lager, datasettfilen du bruker, og eventuelle andre filer som er en del av besvarelsen din, pakkes sammen til én zippet fil som leveres på itsLearning

Innhold

[Oppgave 1 til 6 – 30 % av vektingen 2](#_Toc196209595)

[Oppgave 1 2](#_Toc196209596)

[Oppgave 2 3](#_Toc196209597)

[Oppgave 3 4](#_Toc196209598)

[Oppgave 4 5](#_Toc196209599)

[Oppgave 5 6](#_Toc196209600)

[Oppgave 6 7](#_Toc196209601)

[Oppgave 8 – 30 % av vektingen 8](#_Toc196209602)

[Oppgave 9 - Klasserommet. (40 % av vektingen) 9](#_Toc196209603)

[Del 1: OOP-Klassen Plass 10](#_Toc196209604)

[Del 2: OOP-Klassen Klasserom 11](#_Toc196209605)

[Del 3: Simuleringen 13](#_Toc196209606)

[Oppgave 10 - Last opp dokumentet ditt 14](#_Toc196209607)

# Oppgave 1 til 6 – 30 % av vektingen

## Oppgave 1

Du implementerer abstrakte klasser av objekter i et arv-hierarki. Klasse B arver klasse A. Klasse C arver klasse B. Vi ønsker å kalle objektmetoden *skriv\_ut* for objekter av alle tre klassene, og metoden er implementert i klassene A og C. Hva er korrekt påstand i denne forbindelse (kun ett svar).

1. Klasse B har ikke definert skriv\_ut eksplisitt og objekter av denne klassen vil derfor ikke kunne kalle denne klassemetoden
2. Metoden *skriv\_ut* eksisterer ikke for objekter av klasse B på grunn av *innkapsling*
3. *Assosiasjon* gjør at *skriv\_ut* eksisterer i klasse B, og objektene fra B vil bruke implementasjonen som ligger i klasse C når den ikke er eksplisitt definert i klasse B.
4. *skriv\_ut* finnes for objekter av klasse B siden denne objektmetoden er definert i klasse A og denne er arvet av klasse B.

Svar med kun bokstaven. Eventuelle kommentarer skal i README (.txt eller .md)

Svar:

## Oppgave 2

**Hva er riktig om disse UML-diagrammene?**

Et bilde som inneholder tekst, skjermbilde, Font

KI-generert innhold kan være feil.

1. hus-objekter *assosieres* med rom-objekter på den måten at rom *komponeres (lages)* i rom\_liste
2. hus-objektet *innkapsler* rom-objektene på den måten som fysiske rom er inni hus
3. Rommene som er i *rom\_liste* liste *utvider* hus-klassen (*polymorfi*)
4. Symbolene på relasjons-strekene indikerer assosiasjon i form av *arv* og *abstraksjon*

Svar med kun bokstaven. Eventuelle kommentarer skal i README (.txt eller .md)

Svar:

## Oppgave 3

Flervalg

Gitt følgende lille program, skrevet i pseudokode

SET k TO 10

SET text1 TO «nå er det bare»

SET text2 TO «sekunder igjen!»

WHILE k GREATER THAN -2

DISPLAY text1, k-1, text2

SET k TO k-3

Hvilken eller hvilke påstander er korrekte? Det kan være mer enn ett riktig svar.

1. Dette kan være output:

nå er det bare 10 sekunder igjen

nå er det bare 7 sekunder igjen

nå er det bare 4 sekunder igjen

nå er det bare 1 sekunder igjen

1. Dette kan være output:

nå er det bare 9 sekunder igjen

nå er det bare 6 sekunder igjen

nå er det bare 3 sekunder igjen

nå er det bare 0 sekunder igjen

1. Dette kan være output:

nå er det bare9sekunder igjennå er det bare6sekunder igjennå er det bare3sekunder igjennå er det bare0sekunder igjen

1. Dette kan være output:

nå er det bare9sekunder igjen

nå er det bare6sekunder igjen

nå er det bare3sekunder igjen

nå er det bare0sekunder igjen

nå er det bare-3sekunder igjen

Svar:

## Oppgave 4

Gitt at variabelen a har følgende struktur og innhold:

a = [  
  [10,15,**16**],  
  [40,2,**12**],  
  [**20**,6,24],  
  [10,13,32]  
]

Lag et program som regner ut gjennomsnittet av bare de tallene i variabelen a som oppfyller alle disse kriteriene:

* tallet er større enn eller lik 10
* tallet er mindre enn eller lik 20
* tallet er delelig med 4

Deler av programmet skal med letthet kunne brukes om igjen for andre variabler av samme type (todimensjonale tabeller).

Svar: Lever programfilen.

## Oppgave 5

Ta utgangspunkt i følgende pseudokode for tabell-variabler der hver rad har like mange elementer\*:

SET a TO [  
  [10,15,16],  
  [40,2,12],  
  [20,6,24],  
  [10,13,32]  
]

FUNCTION beregn(d):

SET n TO []

SET t TO []

SET s TO []

FOR hver rad i d

utvid n med 0

utvid t med 0

FOR hver kolonne i rad:

t[siste] = t[siste] + kolonne

INCREMENT n[siste]

ENDFOR

utvid s med t[siste]/n[siste]

ENDFOR

RETURN s

ENDFUNCTION

DISPLAY beregn(a)

1. Forklar med egne ord algoritmen som er beskrevet av pseudokoden ovenfor. Skriv svaret ditt i rammen nedenfor.

|  |
| --- |
| Svar: |

1. Ta utgangspunkt i algoritmen fra punkt a). Gjør nødvendige endringer, og implementer algoritmen til et program som gjør samme type beregninger på hver **kolonne** i variabelen a som pseudokoden beskriver for hver **rad** i variabelen a.

Svar: Lever programfilen.

\* Pseudokode er beskrevet på siste side i eksamensveiledningen.

## Oppgave 6

I denne oppgaven skal du modellere, implementere og teste en objektorientert løsning av følgende problemstilling:

Norgesrekorden for største døgnsum nedbør er 229,6 mm, satt i 1940. (Kilde: yr.no, 2010). Det ble meldt muligheten for 287,8 mm nedbør på et døgn i oktober 2024 under ekstremværet «Jakob» (Kilde: vg.no, 2024), men det slo ikke til.

Harald og Unni krangler om hvem som bor på det våteste stedet. De monterer hver sin nedbørsmåler like ved der de bor, og en mandag begynner de med å registerere døgnsum nedbør en gang i døgnet.

For at ingen skal kunne jukse, så har de samme datamodell:

* en klasse for dagens informasjon, inneholder:
  + ukedag
  + millimeter nedbør
  + setters og getters (metoder for å endre eller hente hver enkelt egenskap i klassen)
* en klasse tar seg av registreringen av alle dagers nedbør, og inneholder:
  + liste med dagens informasjon, og tilhørende getter
  + metode for å registrere nedbør. Harald og Unni trenger bare skrive inn antall millimeter nedbør, klassen tar seg av hvilken ukedag det er (de hopper aldri over en dag)
  + metode for å vise den totale registrerte nedbøren

NB! Du trenger ikke lage meny-programmet som Harald og Unni bruker for å registere data.

1. Lag et UML-diagram for en objektorientert løsning av problemstillingen over.
2. Implementer klassene i ditt programmeringsspråk.
3. Implementer et egnet testprogram for å teste løsningen, og identifiser to mulige feil og ett unntak.\*
4. Implementer nødvendig håndtering av feilene og unntaket du identifiserte i punkt c.\*

Svar: Lever UML-diagrammet, programfilene og testdokumentasjonen.

*\* Det kan være at du allerede i oppgave b har programmert kode som håndterer feil og unntak som kan oppstå i oppgaven. Uansett må du i oppgave c identifisere disse* ***mulige*** *feilene og unntakene for å kunne teste at programmet ditt håndterer dem skikkelig. For oppgave d kan du da identifisere hvilken del av koden som håndterer feilene og unntakene med kommentarer i koden, det bør du jo også gjøre uansett om det er gammel eller ny kode.*

# Oppgave 8 – 30 % av vektingen

Ta for dere datasettet «Innvandring fra 2000» som er vedlagt. Det er hentet fra SSB med tittelen 05183: Innvandrere og norskfødte med innvandrerforeldre, etter landbakgrunn, år og statistikkvariabel fra <https://www.ssb.no/statbank/table/05183/tableViewLayout1/>

Datasettet foreligger i CSV- format og i JSON-format. Det er valgt «Verdensdeler, gammel standard» i «Landbakgrunn» i valget av statistikktabell på forsiden.

Her er en forklaring på overskriftene og strukturen i CSV-filen:

Tittelen på datasettet ligger i første rad. Så følger en blank rad. Deretter kommer overskriftene. Så starter datasettet i rad 4. Det er semikolonseparert. Kodingen er Windows 1252. 

Overskriftene er:

* landbakgrunn
* år
* Innvandrere og norskfødte med innvandrerforeldre

Bli kjent med datasettet og svar på følgende oppgaver

a) Data fra årstallene mellom og inkludert 2020 og 2025 skal fremstilles i en tabell-liknende struktur. Lag et stolpediagram kun for dataene fra 2025.

b) Utvid tabellen fra oppgave a) med årlig endring.

c) Fremstill den totale innvandringen for en landbakgrunn fra datasettet i et linjediagram. Her skal alle årstallene være med. Brukeren skal kunne velge hvilken landbakgrunn som vises i linjediagrammet.

# Oppgave 9 - Klasserommet. (40 % av vektingen)

Et bilde som inneholder skjermbilde, kvadrat, Fargerikt

KI-generert innhold kan være feil.I denne oppgaven skal du utvikle et program som simulerer spredningen av kunnskap i et klasserom.

Graden av kunnskap hos en elev illustreres ved 7 forskjellige nivåer, fra grad 0: ingen kunnskap, til grad 6: særdeles god kunnskap. Disse nivåene skal illustreres med følgende farger:

|  |  |
| --- | --- |
| Kunnskapsnivå | Farge |
| 0.0 - 0.9 | hvit |
| 1.0 - 1.9 | gul |
| 2.0 - 2.9 | oransje |
| 3.0 - 3.9 | grønn |
| 4.0 - 4.9 | blå |
| 5.0 - 5.9 | lilla |
| 6.0 - 6.9 | brun |

Programmet skal utvikles som et objektorientert program med nødvendige klasser og metoder gjennom tre deler.

OBS: Du kan levere separate program fra hver deloppgave, eller bare avsluttende program som inneholder svar på alle tre deler samtidig.

Figur 1: et stykke ut i en tilfeldig time

## Del 1: OOP-Klassen Plass

Vi må forenkle klasseromssituasjonen for å starte modelleringen. Vi ser for oss at klasserommet består av faste sitteplasser for elevene, og gulvplasser hvor læreren kan bevege seg fra plass til plass. Læreren er den eneste som beveger seg i klasserommet i denne modellen.

I denne delen skal du implementere en OOP-klasse som modellerer en plass i klasserommet. OOP-klassen Plass skal ta vare på følgende:

* plassen skal illustreres som en firkant, der klasserommet er et rutenett (se figurene).
* om plassen er en bordplass eller gulvplass. Vi antar at det sitter elever ved alle bordene. Dermed er «bordplass» og «elev» ensbetydende og kan brukes om hverandre.
* om det er en lærer på gulvplassen. Læreren kan ikke være på en bordplass siden de er opptatt med elever. Det er bare en lærer pr klasserom.
* hvilken grad av kunnskap eleven til enhver tid har. Dette vil være et positivt desimaltall under 7. Startverdien er null for alle plasser, og verdien vil øke for elevene etter regler fra oppgavens Del 2.
* økningen i kunnskap siste minuttet med defaultverdi 0, endringer i verdien er beskrevet i Del 2.
* hvor opplagt eleven er til å jobbe og samarbeide, og hvor utholdende eleven er. Dette forenkles til et enkelt positivt heltall (eller prosent) mellom, og inkludert, 0 og 33 (33 %) som settes tilfeldig for hver enkelt elev. Vi kan kalle det drivkraft.
* en metode for illustrasjon etter følgende regler:
  + et bord
    - illustreres med en firkant.
    - Fargen til firkanten er på en gråskala i forhold til verdien på elevens drivkraft, der null drivkraft er svart, og full drivkraft (dvs. verdien 33) er hvit.
      * *Til info: Gråfarger kan f.eks. angis med navn (f.eks. «gray0» er svart, «gray3» er litt lysere, osv. opp til «gray100» som er hvit), eller tupler (f.eks. (0,0,0) er svart, (7,7,7) er litt lysere, osv. opp til (255,255,255) som er hvit. Du kan også bruke andre metoder for å angi gråfargene.*
  + en elev
    - illustreres med en sirkel på bordplassen.
    - Fargen til sirkelen er ut fra graden av kunnskap, etter tabellen gitt over.
  + en gulvplass
    - illustreres med en firkant.
    - fargen på firkanten skal være noe annet enn de fargene som er brukt hittil i oppgaven
  + en lærer
    - illustreres med en sirkel på en gulvplass.
    - fargen er brun.

## Del 2: OOP-Klassen Klasserom

Et bilde som inneholder skjermbilde, kvadrat, tekst

KI-generert innhold kan være feil.I denne delen skal du implementere OOP-klassen Klasserom som skal holde orden på plassene i klasserommet og hvordan graden av kunnskap øker i hver elev.

OOP-klassen Klasserom skal ta vare på følgende:

* opprette alle plassene i klasserommet, både bordplasser og gulvplasser
  + klasserommet skal illustreres som et rutenett på 8x16 ruter
  + bordplassene skal plasseres slik som illustrasjonene viser, som «firerbord» med gulvplass rundt
* starte timen (se figuren) ved å
  + plassere læreren på plassen øverst til venstre til å starte med
  + gi hver elev en tilfeldig heltallsverdi, (eller prosentverdi), for drivkraft, mellom og inkludert 0 og 33 (33 %)
  + gi hver elev grad 0 i kunnskapsnivå

Figur 2: Starten av en tilfeldig time.

*NB! Bildene viser forskjellige simuleringer,   
fargen på «bordet» vil holde seg konstant   
gjennom en simulering.*

* en funksjonalitet for å øke graden av kunnskap i løpet av et minutt på følgende måter:
  + læreren starter minuttet med å gå til en tilfeldig valgt gulvplass i klasserommet. Læreren kan hoppe umiddelbart mellom plasser i klasserommet selv om de er langt fra hverandre og det er bordplasser i veien. Læreren kan bli stående på samme plass over lenger tid dersom den plassen tilfeldigvis ble valgt om igjen.
  + læreren vil ha en eller to elever i umiddelbar nærhet dette minuttet (i de 8 rutene rundt gulvplassen der læreren står vil det være 1 eller 2 bordplasser), disse to elevene vil øke sitt kunnskapsnivå med 0,3 dette minuttet
  + elever som ikke får hjelp av læreren jobber selvstendig og i samarbeid dette minuttet, og lærer av hverandre etter følgende regler:
    1. sannsynligheten for at eleven jobber godt dette minuttet er elevens egen drivkraft, som prosent. Hvis eleven faktisk jobber godt:
       - regn ut gjennomsnittsgraden av kunnskap for alle 4 elevene ved firerbordet
       - ta 5 % av gjennomsnittsgraden
       - legg til 0,2 for egenstudier
       - dette blir elevens økning i kunnskapsnivå for dette minuttet
    2. hvis eleven ikke jobber godt så øker heller ikke kunnskapsnivået
  + Når minuttet er over oppdateres elevenes grad av kunnskap basert på minuttets endringer, bortsett fra for elever som allerede har nådd øverste grad av kunnskap, definert som grad 6.0 eller høyere: de beholder denne graden videre.

## Del 3: Simuleringen

I denne delen skal du implementere en simulering av en skoletime på 90 minutter. Hver oppdatering i simuleringen representerer ett minutt slik minuttet er beskrevet i del 2. Med f.eks. 10 oppdateringer i sekundet vil simuleringen ta ca 9 sekunder totalt.

Når de 90 minuttene er over, skal simuleringen stoppe:

* det gjøres ikke lenger endringer i graden av kunnskap
* fargene på elevene endres ikke lenger
* fargen på læreren settes til å være hvit
* gjennomsnittsgraden av kunnskap i det nye temaet skrives på skjermen uten å dekke over illustrasjonen av bordplassene og elevene.

Et bilde som inneholder skjermbilde, Fargerikt, kvadrat

KI-generert innhold kan være feil.

Figur 3: Slutten av en tilfeldig time.

NB! Bildene viser forskjellige simuleringer,   
fargen på «bordet» vil holde seg konstant   
gjennom en simulering.

# Oppgave 10 - Last opp dokumentet ditt

**Husk å laste opp dokumentet (zip-filen) med svar på oppgave 4, 5b, 6, 8 og 9, og denne word-filen.**

Besvarelsen skal leveres i et komprimert arkiv (en komprimert mappe) som inneholder **alle** filene som er nødvendige for at besvarelsen skal fremstå slik du har ment.

Husk å levere alle vedleggene du bruker i besvarelsen din, i den formen og mappestrukturen du bruker dem. Filene skal ha navn som forteller hvilken oppgave de hører til. Husk også å forklare i en readMe-fil hvordan sensorene skal få vist besvarelsen hvis du har brukt spesielle verktøy eller programversjoner. Dette er spesielt viktig hvis du har brukt et rammeverk som skal installeres.

Der du har brukt en spesiell programvare for å lage dokumentasjon, er det ønskelig at du leverer besvarelsen i et standard dokument som viser dette som tekst og/eller bilde.

Husk å:

1. Samle alle svarene dine i en mappestruktur som er komprimert til en fil.
2. Unngå å levere inn filer som ikke er en del av besvarelsen.
3. Du må bruke et normalt nivå av komprimering (eller bedre) for at filen skal bli liten nok til at den kan leveres inn.

Last opp filen din ved å logge inn på itsLearning i Portalen:

