

# Eksamen IN1010 vår 2022

Skoleeksamen i Silurveien, 4 timer

# IN1010 videre

- [Repetisjonskurs](#) holdes 23.5-3.6 (2 ganger per tema)
- Astro-discourse for spørsmål
- Eksamen 8. juni kl 9:00 og kl 15:00
- Sensur foreligger innen 1. juli
- Løsningsforslag og sensorveiledning legges ut samtidig (eller før)
- Individuell begrunnelse sendes ut til alle (UiO-epost, trenger ikke be om det)
- Utsatt / ny eksamen: Samme form som ordinær, antakelig uken før semesterstart. Bestemmes vanligvis sent (august).

# Emneevaluering

- Fagutvalget ved Ifi (dvs studentene) arrangerer hvert semester emneevaluering.
- Resultatene går til faglærere, utdanningslederne og undervisningsutvalget, et sammendrag publiseres for studenter og ansatte
- Vi faglærere ber dere svare på den. Vi vil veldig gjerne vite deres ærlige mening om emnet IN1010 og undervisningen.
- Du trenger ikke svare på alle punkter – men svarprosent øker kvaliteten

# Hva betyr skoleeksamen i Inspera?

- Møt opp i Silurveien – se studentweb
- all skriving og innlevering skjer i Inspera UNNTATT TEGNINGER
- Inspera gir noe syntaks-hjelp med innrykk og farger – ikke mer
  
- det er ikke tilgang til nett, kompilering eller kjøring av programmer
  
- ingen digitale hjelpemidler kan medbringes
- du kan ha med hva du vil på papir
  
- Tegning gjøres på SPESIALPAPIR (utleveres) og scannes i lokalet.
- Ta med sort/ blå kulepenn (ikke blyant eller andre farger)

# Pensum og læringsmål – fra emnebeskrivelsen

Hva lærer du? Emnebeskrivelse IN1010

[https://www.uio.no/studier/emner/matnat/ifi/IN1010/#learning\\_outcomes](https://www.uio.no/studier/emner/matnat/ifi/IN1010/#learning_outcomes)

Etter å ha tatt IN1010:

- har du god oversikt over programmeringsspråket Java og du kan bruke det til å løse reelle problemer av middels størrelse
- behersker du avanserte objektorienterte mekanismer som subklasser, polymorfi og interface
- har du oversikt over noen grunnleggende datastrukturer (spesielt lenkede lister) og du kan programmere de viktigste operasjonene på dem
- kan du utvikle robuste og pålitelige programmer med godt objektorientert design, og du kan finne alternative løsninger for et gitt problem og vurdere fordeler og ulemper ved de ulike løsningene
- har du kunnskap om parallelle programmer med tråder og du kan benytte dette i enkle programmer
- kjenner du til hendelseshåndtering og kan skrive enkle programmer som håndterer hendelser

# Pensum i Leganto I

- Følg lenke fra semestersiden til [Pensum/litteratur i Leganto](#)
- Alt som er forelest er pensum, se lysark og opptak fra forelesningene
- Alt du har gjort i obliger kan komme på eksamen
- NB! Læringskravet i IN1010 er ikke først og fremst å kunne et pensum, men å kunne skrive gode programmer (rimelig raskt) ved å **bruke programmeringsmetoder som er forelest og som innøves i de forskjellige studieaktivitetene.**

# Pensum i Leganto II

Andre ressurser:

- Big Java kapittel 1-7
  - Kapittel 7.4, kapittel 8 er kjernepensum fra IN1000
  - [Kjernepensum i IN1010: Kapittel 9, 10.1 - 10.3, 13.1- 13.3, 13.5, 16.1, 18, 20.](#)
  - Resten av kapitlene 10 - 18 er kursorisk pensum."
- Notater (lenke fra semestersiden)

# Oppbygging av pensum (fra Timeplanen)

**Mandag 17.1** Introduksjon til IN1010 og til Java

**Tirsdag 18.1** Java: variabler, verdier, objekter og klasser

**Mandag 24.1** Fra Python til Java II

**Tirsdag 25.1** Enhetstesting av objekter, Parameteroverføring

**Tirsdag 1.2** Subklasser I:  
Subklasser og referanser til subklasseobjekter

**Tirsdag 8.2** Subklasser II

**Tirsdag 15.2** Interface (og invarianter i objekter)

**Tirsdag 22.2** Beholdere og generiske klasser I

**Tirsdag 1.3** Beholdere og generiske klasser II

**Tirsdag 8.3** Oppsummering

**Tirsdag 15.3** Programmeringsmønstre

**Tirsdag 22.3** Tråder I

**Tirsdag 29.3** Tråder II

**Tirsdag 5.4** Design og testbarhet

**Tirsdag 19.4** Rekursjon

**Tirsdag 26.4** GUI I

**Tirsdag 3.5** GUI II

*Detaljert pensumliste i  
Leganto, lenket fra  
semestersiden*



# Hva får du på eksamen?

- Programmeringsoppgaver som skal besvares i Inspera
- Evt tegning på papir som scannes inn (datastruktur, klassehierarki)
- Helt sikkert noe fra de grønne temaene
- Nesten helt sikkert ett eller flere blå temaer (med bruk av grønne temaer)
  
- To puljer, uavhengige oppgaver – dvs ikke nødvendigvis samme temaer
- Sammenlignbart omfang og vanskelighetsgrad

# Hvordan vurderer vi besvarelsene?

=> Forståelse og bruk av av sentralt pensum i egne programmer

- Skrivefeil ignoreres vanligvis, men les gjennom det du skriver – feil som kan tyde på manglende forståelse (for eksempel kopiering av kode som ikke passer inn eller svarer på oppgaven) vil gi trekk
- Ignoreres (eksempler):
  - Norske tegn
  - Mangler (noen) ';' , '}' eller '{' om koden ellers gir mening (NB)
  - Småfeil i (gjenkjennbare) navn
  - Blander metodenavn for tilsvarende operasjon i ulike beholdere
- Kan gi (noe) trekk
  - Bruker beholder på en måte som tyder på manglende forståelse
  - Gjennomgående feil/ mangler

# Tidsbruk

Pass tiden – alt inkludert id  
på eventuelle tegninger må  
være klart innen 4 timer

## Bruk tid på å

- få oversikt over hele oppgavesettet
- lese teksten nøye for hver oppgave
- forstå datastrukturen før du programmerer
- vurdere tidsbruk underveis

*Lesbar kode* kan bidra til at sensor forstår hva du mener selv om det f eks mangler en '{'.

Før eksamen: Hva trenger du å ta med *som du rekker å slå opp i*

## Ikke bruk tid på å

- gjenta oppgaveteksten
- skrive kode du ikke blir bedt om (les oppgaven!)
- unødige detaljer i evt. tegneoppgaver (les oppgaven!)
- kommentere (om du ikke trenger å tolke noe du synes er uklart)
  - men angi gjerne underveis i programmet hvilken oppgave du løser
- teste på input om oppgaven ikke ber om det
- testprogrammer om oppgaven ikke ber om det
- velge tilgangsmodifikatorer (public/ private)