

Innlevering på eksamen

- Hvert program kan legges samlet i én .java-fil eller med en klasse per .java-fil.
- Du leverer ikke .class-filer
- Tegninger kan tegnes for hånd (og fotograferes med mobil), eller i et program. Lagres som .jpg, .png eller .pdf-filer. Test ut før eksamen!
- Tegnefiler navngis med oppgaven de besvarer, f.eks. **1a.jpg**

- Oppgavesettet i Inspera vil angi hvor du skal laste opp svar på de ulike oppgavene. Du laster der opp programfilen(e) (og eventuelt tegninger) for den oppgaven det bes om *samlet i en ZIP-fil* .

- Test hvilke programmer du vil bruke under eksamen på forhånd, inkl valg av formater og hvordan du lager og laster opp zip-filer.

Div spørsmål

- oppgavesettet vil kunne lastes ned fra inspera og/ eller eksamenssiden som en pdf
- bruk Mattermost (f eks kanalen town square) for spørsmål **før** eksamen. **Under eksamen vil spørsmål ikke besvares i Mattermost**
 - det er ikke anledning til å kommunisere med andre under eksamen, gjelder også spørsmål/ deling i sosiale medier
 - NB: Enkelte kan ha tilleggstid på eksamen, som derfor ikke betraktes som avsluttet før kl 16.
 - all kommunikasjon med faglærere og teknisk/ administrativ støtte skal skje som angitt på oppgavens førsteside
- Omfang på eksamen sammelignet med prøveeksamen: Begge 6 timer
- Pensum: Se pensumliste. Hva kommer på eksamen: Forelesninger og obliger indikerer hva som er sentralt i pensum

Programmeringsstil, valg av konstruksjoner:

- Må vi sjekke inputverdier? Kun om oppgaven (direkte eller indirekte) angir at dette skal gjøres, ellers antar vi korrekte inndata fra fil/ bruker og som argument til metoder. Kommenter evt om du er i tvil.
- Aksess-modifikatorer: Trenger normalt ikke angi eksplisitt. Ikke bruk tid på dette om det ikke omtales eksplisitt i oppgaven.
- Innkapsling: Ikke get og set på alle instansvariabler (da har du bundet deg til den datastrukturen), men kan være behov for å hente ut (evt beregne) en verdi ved hjelp av en metode. Oftest riktig å unngå direkte aksess av instansvariabler, unntak: Indre klasser, klasser som er tett koblet og alltid brukes sammen.
- while (true) og break: Oftest bedre med en løkketest som viser hva som gjelder i løkken – og etterpå. break kan skjule hva som gjelder etter løkken. Kan føre til uventede feil om du endrer koden utenfor løkken.