# Oppgaveoversikt og studietips

## Tidligere eksamensoppgaver

Du får tilgang til tidligere eksamensoppgaver via denne linken: Tidligere eksamensoppgaver i IN1010. Merk at det har vært eksamener av ulik lengde og ulik type, som har implikasjoner for hva som skal gjøres og hvor mye det er å gjøre. Eksamenen du skal ha i IN1010 er en skriftlig skoleeksamen på fire timer i Silurveien, der du kan ha med deg alt av skriftlige hjelpemidler. Selv om du skal ha en skriftlig skoleeksamen på fire timer, kan det likevel være god trening å gjøre eksamenssett for andre eksamenstyper i tillegg.

***Studietips:*** *Ved å gjøre eksamenssett får du en indikasjon på hva du kan veldig godt og hva du bør forberede deg mer på. Du får også øvd deg på selve eksamenssituasjonen. Sett av fire timer, skru av all hjelp i Visual Studio Code, eller bruk et annet tekstredigeringsverktøy. Husk at du bare får veldig enkel highlighting av litt syntaks i Inspera - ingen annen hjelp - og du kan hverken kompilere eller kjøre programmet ditt. Derfor er det lurt å øve seg på forhånd, på prøveeksamen og flere eksamenssett. Sitt gjerne sammen med andre, lat som det faktisk er eksamen og unngå å snakke sammen på fire timer. Deretter kan dere diskutere oppgavene og løsningene med hverandre - det er veldig lærerikt!*



## Tidligere ukeoppgaver og Trix-oppgaver

Det finnes ukeoppgaver tilhørende alle gruppetimene som har vært i IN1010 denne våren. Du får tilgang til dem ved å følge denne linken: Gruppeoversikt i IN1010, og så velge gruppe. Trix-oppgavene finner du her: Trix i IN1010. Oppgavesettene inneholder både mindre avanserte og mer utfordrende oppgaver. Enkelte av dem går rett i kjernen av det som er gjennomgått på forelesning, mens andre lar deg utforske mer innen et gitt IN1010-tema. Hvis du løser en eller flere eksamensoppgaver for å få en indikasjon på hva du kan, er det fint å blant annet bruke tidligere ukeoppgaver og Trix-oppgaver til å videreutvikle deg. Du kan også fokusere på innleveringsoppgavene, for eksempel de du ikke har prøvd deg på.

***Studietips:*** *Forsøk å løse en hel oppgave før du eventuelt ser på et løsningsforslag. Et løsningsforslag er nettopp bare et forslag, og det er ingen absolutt fasit, fordi oppgaver i programmering ofte kan løses på så mange måter. Hvis du bruker et løsningsforslag, øv deg på å forklare likheter og forskjeller mellom din egen løsning og forslagets løsning. Husk alltid å forstå hva som skjer og hvorfor det skjer. Øv deg gjerne på "algoritmisk tenkning"; det er en måte å tenke på der du bryter ned en større oppgave i mindre, mer håndterlige deler, før du setter delene sammen til en helhetlig løsning. Gå ut fra at logikken din er riktig før du prøver å kompilere og kjøre, fremfor å prøve og feile blindt uten å skjønne hva som skjer. Løs gjerne oppgavene sammen med parprogrammering. Utfordre deg - individuelt og/eller sammen med andre - på å lage egne oppgaveutvidelser.*