# Informasjonsteknologi 2 – dette må du mestre

11.10.26

**Laget av KI: snakk med læreren din først**

Denne oversikten er laget av KI ut fra tidligere eksamensoppgaver. Den kan ta feil, og den kjenner ikke deg. Læreren din vet best hva *du* bør øve på, så spør læreren først.

Står du helt på bar bakke, eller leser du til privatisteksamen, kan du bruke arket som et utgangspunkt for øvingen.

Denne siden bygger på alle de sju eksamenene i Informasjonsteknologi 2 etter den nye læreplanen, fra våren 2023 til våren 2026. Vi har sortert oppgavene i oppgavetyper og telt hvor ofte hver type har vært med.

- **Karakter 2:** Forbered deg på alt i de grønne boksene. Det er de lette delene av oppgavetypene som er med på (nesten) *hver* eksamen. Du trenger ikke få til alt for å få 2, men jo mer av det du kan, jo tryggere er du.
- **Karakter 4:** Forbered deg i tillegg på de blå boksene, og én oppgavetype som bare står under karakter 4. Heller ikke her må du kunne alt for å få 4, men har du forberedt deg på alt, kan du også få 5 eller 6.
- **Øv på:** Under hver oppgavetype står eksamensoppgaver å øve på. QR-koden legger alle eksamensoppgavene av typen inn i *Mine oppgaver* på matematikkoppgaver.no.

**Eksamensordningen fra våren 2027**

Fra våren 2027 mister matematikk, Fysikk 2, Kjemi 2 og Biologi 2 datamaskinen på eksamen. IT 2 er ikke nevnt i den endringen, så vi regner med at eksamen fortsatt er heldigital: 5 timer på din egen datamaskin, med alle hjelpemidler som ikke krever internett, men uten KI. Det er ikke bekreftet ennå, så sjekk med læreren din. Fordi du fortsatt har datamaskin, er ingen oppgaver tatt ut av øvelistene.

## Oversikt

| Oppgavetype | Seksjon | Med på | Karakter 2 | Karakter 4 |
|---|---|---|---|---|
| Lese, spore og sortere pseudokode | 1 | 7 av 7 | ✓ | ✓ |
| OOP-begreper og UML-relasjoner | 1 | 6 av 7 | ✓ | ✓ |
| Forklare og implementere en algoritme | 1 og 2 | 7 av 7 | ✓ | ✓ |
| Klasser etter klassediagram | 2 | 5 av 7 (alle fra V24) | ✓ | ✓ |
| Drøfte et etisk dilemma | 3 | 7 av 7 | ✓ | ✓ |
| Analysere et datasett | 4 | 7 av 7 | ✓ | ✓ |
| Spill og simulering | 5 | 7 av 7 | ✓ | ✓ |
| Flytskjema | 1 | 4 av 7 | | ✓ |

## Slik er eksamen

| Eksamen | Hjelpemidler (antatt uendret fra våren 2027) |
|---|---|
| Hele eksamen, 5 timer | Alle hjelpemidler som ikke krever internett, og noen utvalgte nettressurser. Ikke kommunikasjon og ikke KI. |

Eksamen er ikke delt i Del 1 og Del 2. Du svarer på noen oppgaver direkte i eksamenssystemet, og leverer programmer og diagrammer i én zip-fil til slutt. Settet er delt i seksjoner med anbefalt tid: korte oppgaver (rundt 1 time), en klasseoppgave, en drøfting (30 minutter), et datasett (1 time) og et stort spill eller en simulering (1,5–1,75 timer). I V24 og V25 ga eksamen 60 poeng. De korte oppgavene før klasseoppgaven ga 9,5 poeng i V24 og 12 poeng i V25, og den store siste oppgaven ga 24 og 18 poeng.

## Øv med hele eksamenssett

Det viktigste rådet: Gjør hele eksamenssett på tid, *uten* mer hjelp enn du har lov til på eksamen. Ingen løsningsforslag, ingen nettsøk og ingen KI før du er ferdig.

| Når | Gjør dette |
|---|---|
| 14 dager før eksamen | Ett helt eksamenssett på tid |
| 7 dager før eksamen | Et nytt eksamenssett |
| 2 dager før eksamen | Et siste eksamenssett |

Skriv ned hvor mange poeng du fikk til uten hjelp hver gang, og se etterpå på oppgavene du ikke klarte. Jobber du jevnt og trutt mellom settene, får du til mer og mer. Bruk oppgavetypene under til å øve på det du ikke fikk til.

# Korte oppgaver og klasseoppgaven

## Lese, spore og sortere pseudokode

**Slik ser det ut**

*Seksjon 1 · med på 7 av 7 · ofte to eller tre små oppgaver på 1–2 poeng hver*

En for-løkke starter med `b = 4` og `c = 5`. For hver runde vises `c`, så telles `b` ned med 1 og legges til `c`. Hva vises? Svaret er `5 8 10` (V24). Eller: Sorter ti stokkede pseudokodelinjer til et program som teller hvor mange av fem tall som er større enn 10 (V26).

Alle oppgavene om å lese og sortere pseudokode (13)

**Karakter 2**

- Spore en løkke for hånd: Lag en tabell med én kolonne per variabel, og skriv ned verdiene etter hver runde.
- Vite forskjellen på løkkene: `FOR` når du vet hvor mange ganger, `WHILE` så lenge en betingelse er sann.
- Sortere linjer: Finn først `SET … TO`-linjene som gir startverdier, så løkka, så `IF`-testen inni, og til slutt `DISPLAY` etter løkka.
- Kjenne nøkkelordene i UDIRs pseudokode: `SET … TO`, `READ`, `DISPLAY`, `IF … ELSE … ENDIF`, `WHILE … ENDWHILE`, `FOR … ENDFOR`.
- Med `IF … ELSE IF … ELSE` testes betingelsene ovenfra og ned, og bare den første som er sann, blir utført.

**Øv på:** Løkke med nedtelling og løpende sum i pseudokode, Påstander om for- og while-løkker, Sortere pseudokode som teller tall større enn 10, Sortere pseudokode for temperaturmeldinger

**Karakter 4: i tillegg**

- Spore en funksjon med `IF … ELSE IF` og modulo: `i % 2 EQUAL TO 0` betyr at `i` er et partall. Sjekk den strengeste betingelsen først (delelig med både 2 og 3).
- Se når en løkke aldri kjører: Starter `i` på 1 og betingelsen er `i > m` med `m = 3`, blir det ingen utskrift.
- Sortere lange programmer med nøstede `IF`-tester (V24 hadde 18 linjer).
- Skrive den manglende delen av en funksjon slik at den gir et oppgitt resultat.

**Øv på:** TrommeLom-funksjon i pseudokode, Funksjon med vekslende sum i pseudokode, Antall utskrifter fra en while-løkke, Sortere pseudokode for fortegn og partall, Fullføre pseudokode med dyrenavn

## OOP-begreper og UML-relasjoner

**Slik ser det ut**

*Seksjon 1 · med på 6 av 7 · to eller tre flervalgsoppgaver på 1–1,5 poeng hver*

«Hva er hovedformålet med en konstruktør i objektorientert programmering?» (V26) eller «Hvilket prinsipp gjør at metoder kan ha samme navn i superklassen og subklassene og likevel kalles riktig?» (V25, svaret er polymorfisme).

Alle oppgavene om OOP-begreper (12)

**Karakter 2**

- Klasse: en mal for objekter. Objekt: én instans laget fra klassen.
- Konstruktør: setter startverdiene til attributtene når objektet lages.
- Arv: en subklasse arver attributter og metoder fra en superklasse. B arver fra A betyr at B er en *spesialisering* av A.
- Polymorfisme: samme metodenavn gjør forskjellige ting i ulike subklasser.

**Øv på:** Hovedformålet med en konstruktør, Hva en klasse er i objektorientert programmering, Mekanisme for å generalisere klasser til en superklasse, OOP-prinsipp for metoder med samme navn i superklasse og subklasser

**Karakter 4: i tillegg**

- Skille relasjonene i UML: assosiasjon (vanlig strek), aggregering (hul rombe, delen kan finnes alene), komposisjon (fylt rombe, delen finnes bare inni helheten) og arv (hul trekant).
- Innkapsling (`-` privat, `+` offentlig) og abstraksjon (skille grensesnittet fra de interne detaljene).
- Lese multiplisitet som `0..1` og `0..*`, og begrunne hvilken relasjon du velger.

**Øv på:** Relasjon når objekter bare finnes inne i en annen klasse, Relasjon når et felt inneholder objekter av en annen klasse, Relasjon mellom student og fakultet, OOP-prinsipp for å skille grensesnitt og interne detaljer, Arv i klassediagram for transportsystem

## Forklare og implementere en algoritme

**Slik ser det ut**

*Seksjon 1 eller 2 · med på 7 av 7 · 4,5–6 poeng (V24 og V25), ofte a) forklar og b) implementer*

Du får pseudokoden til en kjent algoritme: Fibonaccitall (V24), palindromtall (H24), bokstavforskyvning (V25) eller Kaprekars rutine (H25). For 3142 gir Kaprekars rutine 4321 − 1234 = 3087, så 8730 − 0378 = 8352, så 8532 − 2358 = 6174, og deretter står tallet stille.

Alle algoritmeoppgavene (9)

**Karakter 2**

- Kjøre algoritmen for hånd med ett eksempel før du forklarer den. Implementer og kjør koden for å sjekke svaret, særlig om løkka går «til» eller «til og med» siste indeks.
- Implementere kort pseudokode linje for linje i ditt eget språk, og kjøre programmet.
- Kjøre en testplan: Fyll inn faktisk resultat ved siden av forventet resultat for hver test.

**Øv på:** Boblesortering i pseudokode (a), Testplan for karakterprogram med poengsum (a), Fibonaccitall i pseudokode (a)

**Karakter 4: i tillegg**

- Forklare hva algoritmen gjør *som helhet*, ikke linje for linje. Sensor gir ikke poeng for å gjenta pseudokoden med ord.
- Utvide algoritmen: telle alle tresifrede palindromtall, eller vise de ti første partallene i Fibonaccifølgen.
- Finne logiske feil med grenseverdier: Testen 50 skal gi «Bestått», men programmet sier «Ikke gyldig poengsum!» fordi det bruker `GREATER THAN 50` der det skulle stått «større enn eller lik». Og −1 gir «Ikke bestått» fordi den nedre grensen mangler.
- Vurdere to riktige løsninger mot hverandre: Hvilken går gjennom lista færrest ganger?

**Øv på:** Bokstavforskyvning i pseudokode, Kaprekars rutine for firesifrede tall, Palindromtall i pseudokode, Finne det nest største tallet i en liste, Summen av de ti første trekanttallene i pseudokode

## Klasser etter klassediagram

**Slik ser det ut**

*Seksjon 2 · med på 5 av 7, men på alle sett fra V24 · 5,5–12 poeng, nesten alltid a) diagram, b) implementer, c) test, d) feil og unntak*

Lag klassene `Bok` og `Låner` med utlån og tilbakelevering, et testprogram som sjekker at en utlånt bok ikke kan lånes ut igjen, og håndtering av feil og unntak (V25, 3 poeng per delpunkt).

Alle klasseoppgavene (5)

**Karakter 2**

- Implementere klasser fra et ferdig klassediagram: én klasse per boks, attributtene i konstruktøren og én metode per linje.
- Lage et testprogram som oppretter objekter og kaller alle metodene, med utskrift av resultatet.
- Tegne et enkelt klassediagram med navn, attributter med datatype og metoder.

**Øv på:** Kalkulatorklasse med fire regneoperasjoner (a, b og c), Batteriklasse med lading og energibruk (a)

**Karakter 4: i tillegg**

- Håndtere feil og unntak: deling med null, feil datatype fra input, verdier utenfor lovlig område (lade over kapasiteten, låne en bok som er utlånt). Bruk `try`/`except` (eller `try`/`catch`) og valider i metodene.
- Forklare relasjonen i diagrammet: En `Digitalklokke` *består av* to `Tallviser`-objekter (komposisjon).
- Bruke objekter inni objekter: en liste med lånte bøker, eller et togsett med lokomotiv og vogner.
- Utvide modellen med en superklasse og gi hver togdel et unikt serienummer med en klassevariabel som teller.

**Øv på:** Bibliotek med bøker og lånere, Digitalklokke med tallvisere, Togsett med lokomotiv og vogner

# Drøfting, datasett og den store oppgaven

## Drøfte et etisk dilemma

**Slik ser det ut**

*Seksjon 3 · med på 7 av 7 · 4–6 poeng, 30 minutter*

Noah (15) går på en skole der kameraer filmer klasserommet og kantina. Drøft et etisk dilemma som kan oppstå (V26). Andre case: VR-behandling av sosial angst (V25) og en KI-chatbot som samtalepartner (H25).

Alle drøftingsoppgavene (7)

**Karakter 2**

- Si tydelig hvilket dilemma du drøfter, med to verdier som står mot hverandre: trygghet mot personvern, eller hjelp mot autonomi.
- Gi minst ett argument for og ett mot, og knytt dem til personene i caset.
- Avslutte med din egen mening og en begrunnelse.

**Øv på:** Etisk dilemma ved kameraovervåking i skolen, Etisk dilemma ved KI-chatbot som samtalepartner

**Karakter 4: i tillegg**

- Bruke fagbegreper: personvern, personopplysninger, samtykke og informert samtykke, dataminimering, formålsbegrensning, autonomi, ansvar og tillit.
- Se konsekvenser både for den ene personen og for samfunnet.
- Veie argumentene mot hverandre før du konkluderer, ikke bare liste dem opp.
- Oppgi kilder hvis du bruker noe du har lest.

**Øv på:** Etiske dilemmaer ved VR-behandling av sosial angst, Etiske dilemmaer ved velferdsteknologi for Kari, Deepfake-svindel og deepfakes i jobbintervjuer

## Analysere et datasett

**Slik ser det ut**

*Seksjon 4 · med på 7 av 7 · 12–15 poeng, to eller tre delpunkter*

Du får et datasett i både CSV og JSON. a) Les inn dataene og vis summer i en tabell. b) La brukeren velge noe (et fylke, et år, et fagområde) og vis sortert resultat med antall og prosent. c) Lag et diagram (V25: friluftsaktiviteter per fylke).

Alle datasettoppgavene (7)

**Karakter 2**

- Lese inn en CSV- eller JSON-fil med riktig tegnsett (UTF-8 eller Windows-1252) til en liste med ordbøker eller objekter.
- Gjøre tekst om til tall før du regner.
- Summere per kategori og skrive resultatet ut som en ryddig tabell.
- Legge datasettfila i samme mappe som programmet og bruke relativ sti, så programmet virker når sensor kjører det.

**Øv på:** Deltakelse i friluftsaktiviteter per fylke (a), Tidsbruk på ulike aktiviteter etter kjønn (a), Elevtall i skolefag over tre år (a)

**Karakter 4: i tillegg**

- Filtrere på et valg brukeren gjør, og sortere synkende eller stigende.
- Regne prosentandel og prosentvis endring: fra 200 til 250 elever er en oppgang på $\frac{250 - 200}{200} \cdot 100\,\% = 25\,\%$.
- Lage stolpe-, linje- eller sektordiagram, og velge topp 3 eller topp 10.
- Vaske data: tall lagret som tekst med `+` og komma, manglende verdier og ukjent tegnsett. Skriv en kommentar om hva du har vasket.
- Hente kategoriene fra datasettet, ikke skrive dem inn for hånd. Hardkodede verdier gir lite uttelling.

**Øv på:** Registrerte kjøretøy og elbilandel per region, Befolkningsutvikling i Norge, YouTube-kanaler per land, Oversikt over apper i Google Play Store

## Spill og simulering

**Slik ser det ut**

*Seksjon 5 · med på 7 av 7 · 18–24 poeng, 1,5–1,75 timer*

Noen ganger et spill, andre ganger en simulering i et rutenett: PacTroll (V23), Manic Mansion (H23), Livets spill (V24), virusspredning (H24), skogbrann (V25), bunndyr og plankton (H25) og stolpebyggeren (V26). Alt skal løses objektorientert.

Alle spill- og simuleringsoppgavene (7)

**Karakter 2**

- Få opp et vindu eller et rutenett i riktig størrelse, og tegne objektene i det.
- Lage én klasse per ting i spillet (spiller, hindring, celle) med posisjon og en metode for å tegne seg selv.
- Styre spilleren med piltastene, eller gi hver celle en tilfeldig starttilstand (Livets spill: $\frac{1}{3}$ sjanse for levende).
- Lever det du har, selv om spillet ikke er ferdig. Poengene deles ut for hver del som virker: I V24 ga de to første delpunktene 8 av 24 poeng.

**Øv på:** Livets spill (a og b1), Spillet PacTroll (b, bare vindu, objektene tegnet og spilleren styrt med piltastene)

**Karakter 4: i tillegg**

- Oppdage kollisjoner mellom rektangler, og plassere nye objekter tilfeldig uten overlapp.
- Telle naboene til en celle (fire eller åtte), og passe på kantene av rutenettet.
- Oppdatere alle cellene *samlet*: Regn ut neste generasjon fra den gamle før du bytter.
- Holde styr på poeng, liv og tilstander (frisk, smittet, syk, immun), og vise en sluttskjerm.
- Forklare og tilpasse et ferdig klassediagram for spillet.

**Øv på:** Simulering av skogbrann, Simulering av virusspredning i en populasjon, Simulering av bunndyr som spiser plankton, Spillet stolpebyggeren, Spillet Manic Mansion

# Bare for karakter 4

## Flytskjema

**Slik ser det ut**

*Seksjon 1 · med på 4 av 7 · lese eller tegne et flytskjema*

Hvilken av tre pseudokoder gjør det samme som flytskjemaet? Flytskjemaet summerer partallene fra 1 til et innlest tall, så 10 gir 2 + 4 + 6 + 8 + 10 = 30 (V26). Eller: Lag et flytskjema for billettpris etter alder (H23).

Alle flytskjemaoppgavene (4)

**Karakter 4**

- Kjenne symbolene: avrundet boks for start og slutt, rektangel for en handling, rombe for et valg og parallellogram for inn- og utdata.
- Følge pilene med et lite tall og sammenligne med hver pseudokode, særlig startverdien og betingelsen i løkka.
- Tegne et flytskjema med flere valg etter hverandre (alder 15 eller yngre, 67 eller eldre, ellers voksen), og lagre det som pdf eller bilde.

**Øv på:** Flytskjema og pseudokode for sum av partall, Flytdiagram for billettpris etter alder, Pseudokode som samsvarer med et flytskjema

## Dette kan du hoppe over

Disse har vært med, men er borte fra eksamen, ikke testet eller så vanskelige at tiden er bedre brukt på oppgavetypene over:

- Flervalg om en forberedelsestekst (fram til H24). Eksamen har ikke hatt forberedelsesdel siden V25, og casen står nå i selve oppgaven.
- Systemutvikling med Scrum og Git. Det står i læreplanen, men har ikke vært på skriftlig eksamen.
- Nettsider, animasjon og video fra eksamenene før 2023. Det hører til den gamle læreplanen.
- De siste og vanskeligste delene av den store oppgaven, som stabling og rotasjon i stolpebyggeren (V26) eller diagrammer som oppdateres mens brukeren endrer filteret (V24).