

Kort om kursene IN1900, MAT-IN1105, IN-KJM1900

Joakim Sundnes¹ Ole Christian Lingjærde¹

Department of Informatics, University of Oslo¹

Aug 23, 2017

- Info om kursene IN1900, MAT-IN1105, IN-KJM1900
 - Generell info
 - Kort gjennomgang av info på semestersidene
 - Nyttige tips om programvare og arbeidsform
- Første introduksjon til programmering
 - Live-programmering; "Hello world!"
 - Programmering av matematiske formler

IN1900 er en første introduksjon til å programmere datamaskiner

- Programmering er *svært* viktig i industri og forskning!
- Programmering vil bli brukt i veldig mange senere emner - derfor er IN1900 et sentralt kurs
- Hvorfor?
Programmeringen gjør matematikken mye mer anvendbar
- Tre perspektiver på matematikk i høst:
 - tradisjonell kalkulus (MAT1100/MAT1001)
 - numerisk (datamaskinvennlig) matematikk (MAT-INF1100)
 - programmering av numerisk matematikk (IN1900/MAT-INF1105/IN-KJM1900)

MAT-IN1105 = IN1900 uke 1-6 + MAT-INF1100

- Uke 34: Følger forelesninger i INF1100 og MAT-INF1100
- Uke 35-40: Sammenfallende undervisning med INF1100:
 - forelesninger: uke 34-39
 - gruppeøvelser (obliger): uke 35-40
- Samme midtveiseksamen som IN1900
- Fullstendig sammenfallende undervisning med MAT-INF1100 etter det

IN-KJM1900 = IN1900 uke 1-8 + prosjekt

- Fullstendig sammenfallende undervisning og obliger med IN1900:
 - forelesninger: uke 34-41
 - gruppeøvelser (obliger): uke 35-42
- Samme midtveiseksamen som IN1900
- Eget kjemi-relatert programmeringsprosjekt etter det

- IN1900: <http://www.uio.no/studier/emner/matnat/ifi/IN1900/h17>
- MAT-IN1105: <http://www.uio.no/studier/emner/matnat/math/MAT-IN1105/h17>
- IN-KJM1900: <http://www.uio.no/studier/emner/matnat/ifi/IN-KJM1900/h17>
- Se spesielt IN1900 undervisningsplan for info om hva som skjer hver uke

Kursene krever ingen forkunnskaper om programmering

- Vi starter med veldig enkle programmer, men går relativt fort fram
- For mange er programmeringen en helt ny måte å bruke en datamaskin på
- En del nye verktøy; editor, terminalvindu, etc
- Verktøy må læres underveis; kom tidlig i gang, spør om hjelp ved behov
- Har du aldri programmert før? Kom tidlig i gang, prioriter de første ukene.
- Har du programmert Python tidligere? Følg med på hva som skjer i undervisningen likevel

Python bakgrunnssjekk

Undervisningen består av øvelser og forelesninger

- Plenumsundervisning onsdager 10.15-12.00 og fredager 12.15-14.00 i Sophus Lies auditorium
- 1. time: oppgaver fra forrige forelesningstime løses i plenum
- 2. time: forelesning av nytt stoff
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Delta på *alle* undervisningstimene!

Undervisningen består av øvelser og forelesninger

- Plenumsundervisning onsdager 10.15-12.00 og fredager 12.15-14.00 i Sophus Lies auditorium
- 1. time: oppgaver fra forrige forelesningstime løses i plenum
- 2. time: forelesning av nytt stoff
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Delta på *alle* undervisningstimenel

Undervisningen består av øvelser og forelesninger

- Plenumsundervisning onsdager 10.15-12.00 og fredager 12.15-14.00 i Sophus Lies auditorium
- 1. time: oppgaver fra forrige forelesningstime løses i plenum
- 2. time: forelesning av nytt stoff
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Delta på *alle* undervisningstimene!

Undervisningen består av øvelser og forelesninger

- Plenumsundervisning onsdager 10.15-12.00 og fredager 12.15-14.00 i Sophus Lies auditorium
- 1. time: oppgaver fra forrige forelesningstime løses i plenum
- 2. time: forelesning av nytt stoff
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Delta på *alle* undervisningstimene!

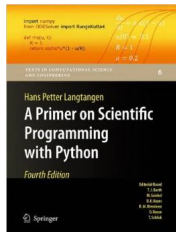
Undervisningen består av øvelser og forelesninger

- Plenumsundervisning onsdager 10.15-12.00 og fredager 12.15-14.00 i Sophus Lies auditorium
- 1. time: oppgaver fra forrige forelesningstime løses i plenum
- 2. time: forelesning av nytt stoff
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Delta på *alle* undervisningstimene!

Undervisningen består av øvelser og forelesninger

- Plenumsundervisning onsdager 10.15-12.00 og fredager 12.15-14.00 i Sophus Lies auditorium
- 1. time: oppgaver fra forrige forelesningstime løses i plenum
- 2. time: forelesning av nytt stoff
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Delta på *alle* undervisningstimene!

- Lærebok skrevet spesielt for IN1900/INF1100
- Vi bruker 5. utgave i høst
- 4. utgave er mulig å bruke, men nummerering av oppgaver og kapitler er forskjellig. 3. utgave og tidligere anbefales ikke
- Vi går gjennom ca ett kapittel pr uke, stort sett med ett til to nye sentrale begreper



Det kreves innlevering av 3-5 obligatoriske oppgaver hver uke

- “Løp 1”: Mange små obligatoriske oppgaver
 - 3-5 obligatoriske oppgaver hver uke (vurderes til bestått eller ikke bestått)
 - De fleste oppgavene teller 1 poeng
 - Krav IN1900: 45 av 70 poeng før 1. des.
 - Krav MAT-IN1105: 17 (av 26) p fra oppgavene i uke 35-39, men det blir gitt ekstraoppgaver etter midtveiseksamen
 - IN-KJM1900: 24 (av 37) p fra oppgavene i uke 35-41, pluss prosjektoppgave
- “Løp 2”: Færre, men større obligatoriske oppgaver
 - Passer for dere med god programmeringserfaring
- Eksamen:
 - Midtveiseksamen 10. oktober - teller 25% av karakteren
 - Avsluttende eksamen 18/12 - teller 75% av karakteren

Hvordan du må jobbe

- Foran hver forelesning må du ha lest ukens kapittel i læreboken
- Foran hver oppgaveløsning i plenum må du selv ha forsøkt å løse oppgavene (les kapittelet først!)
- Etterarbeid oppgavene når du har sett løsning i plenum
- Nå er du klar for ukens obliger: Du kan gjøre dem på terminalstue under veiledning

Merk:

- Spesielt forelesningene går frem mye fortere enn klasseromsundervisningen i videregående skole
- Undervisningen forutsetter at du er forberedt og at du kan forrige ukes temaer

Hvordan du må jobbe

- Foran hver forelesning må du ha lest ukens kapittel i læreboken
- Foran hver oppgaveløsning i plenum må du selv ha forsøkt å løse oppgavene (les kapittelet først!)
- Etterarbeid oppgavene når du har sett løsning i plenum
- Nå er du klar for ukens obliger: Du kan gjøre dem på terminalstue under veiledning

Merk:

- Spesielt forelesningene går frem mye fortere enn klasseromsundervisningen i videregående skole
- Undervisningen forutsetter at du er forberedt og at du kan forrige ukes temaer

Har du spørsmål om ukeoppgaver, eksempler fra forelesninger eller programmering generelt?

- Post dem på piazza!
- Alle skal ha blitt invitert til et eget IN1900-forum (link fra emnesiden)
- Gruppelærere, studenter (og delvis forelesere) svarer aktivt
- Gjennomsnittlig responstid høsten 2016; 20 minutter

Du må lære programmering ved å programmere mye

- Du kan ikke lese deg til programmering
- De fleste synes programmering er krevende i begynnelsen - så blir det utrolig gøy!
- Oppskrift på suksess: vær godt forberedt til undervisningen - det gir deg mest fritid og mest læring
- Forventet arbeid er 13 timer med IN1900 hver uke (6 t undervisning, 7 t selvstudium)

Hvor mye matematikk må jeg kunne på forhånd?

- Nesten alle eksemplene i IN1900 handler om bruk av matematikk
- Vi bygger (i prinsippet) på R2 fra vgs
- Men matematikken i IN1900 er stort sett *numerisk* matematikk (MAT-INF1100)
- Vi håper at IN1900 skal belyse matematikk fra en ny vinkel og hjelpe deg til å forstå matematikk bedre samtidig som du lærer å programmere

Alt undervisningsmateriale er på engelsk

- Muntlig undervisning foregår på norsk
- Alt skriftlig materiale er på engelsk
- Hvorfor?
- Det mangler gode norske ord for mange ord/uttrykk i programmering
- Du finner mye informasjon om programmering på nettet og i bøker - nesten all denne informasjonen er på engelsk og da må du kunne de engelske uttrykkene
- Mesteparten av undervisningsmateriellet på UiO er på engelsk
- I jobbsammenheng kan du regne med at alt skriftlig foregår på engelsk
- Boken og undervisningsmaterialet brukes ved mange utenlandske universiteter

Bruk av egen laptop, operativsystem, etc

- Du kan fint gjennomføre kurset uten egen laptop, alle oppgaver kan gjøres på terminalstuene
- Du kan også bruke egen laptop; Python kjører på Windows, Mac og Linux
- Anbefalt arbeidsform; teksteditor + terminalvindu
- Se emnesidene for tips om programvare
- Laptophjelpen tilbyr hjelp til installering i starten av semesteret
- Hvis du har problemer med bruk av egen laptop, bruk terminalstuene til å komme i gang med programmering og oppgaver. Løs problemene senere.
- Spør om hjelp på gruppetimene

Viktig info om UiO-brukernavn

- Du trenger UiO-brukernavn og passord for å:
 - Logge inn på maskiner på terminalstuene
 - Logge inn på UiO-maskiner hjemmefra
 - Levere oppgaver i [devilry](#)
- Brukernavn og passord aktiveres når du har betalt semesteravgiften!
- Din UiO-bruker er også en mail-adresse (`brukernavn@student.matnat.uio.no`). Sjekk denne jevnlig!

Python 2 vs Python 3

- To Python-versjoner brukes mest i dag: 2.7 og 3.6
 - Python 3.6 er det vanligste i generell programmering
 - Python 2.7 er fortsatt mye brukt i matematikk og vitenskapelig programmering
- Foiler, oppgaver i plenum etc kommer til å være Python 3.6 (nytt for 2017)
- Eksempler i læreboka er Python 2.7
- For IN1900 er forskjellen veldig liten (se beskrivelse på emnesidene)
- Det finnes programmer for å konvertere mellom Python 2 og 3 (se læreboka)
- Mer info kommer senere

- Ikke overraskende er google et meget nyttig verktøy også for å lære programmering, med noen ulemper:
 - Du må vite hva du skal søke etter, dvs du må kunne en del Python for å søke effektivt
 - Mye informasjon er veldig teknisk og lite egnet for nybegynnere; krever litt trening å lese
- Tips: kryptiske feilmeldinger fra Python kan limes rett inn i google søkefelt. Gir ofte (men ikke alltid) en grei forklaring på hva som er galt
- NB: Tidligere INF1100-studenter (og andre) har lagt ut løsninger på oppgaver på web. Vær *meget* forsiktig med å bruke disse

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
 - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
 - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
 - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.
- Kopiere eller skrive av løsningsforslag?
 - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

Samarbeid om obligatoriske oppgaver

- Samarbeid er bra, men pass på at alle bidrar, og at alle forstår de løsningene man kommer fram til
- Alle skriver sin egen kode, og leverer sin egen innlevering.
- Skriv en kommentar i innleveringen om hvem du har samarbeidet med:
 - Ved samarbeid blir koden gjerne veldig lik
 - IFI har verktøy som finner like innleveringer, og flagger dette som mulig avskrift/fusk
 - Veldig enkelt å rydde opp i hvis vi vet hvem som har samarbeidet med hvem. Tar fort mye tid ellers.

Tilbakemelding på undervisningen

- Still spørsmål underveis (ikke lett i Sophus Lie...)
- Kom med synspunkter etter forelesning eller send epost
- Hva fungerer bra, hva er mindre bra
- Er noen tema spesielt vanskelige?
- Er info på web utdatert eller feil?
- Alle blir hørt, selv om ikke alle forslag lar seg gjennomføre

Viktig hjemmearbeid første uke

- Les gjennom informasjon på semestersidene:
 - Sjekk ut alle lenker fra hovedsiden
 - Bli kjent med undervisningsplanen
- Finn en terminalstue med Linux-maskiner:
 - Logg inn med UiO-brukernavn
 - Åpne en editor og et terminalvindu
 - Skriv `python` i terminalvinduet og se hvilken versjon du kjører (øverste linje, sannsynligvis 3.6 eller 2.7). Avslutt Python ved å skrive `quit()` eller med *Ctrl-D*.
- Test eventuelt installasjon på egen maskin (se info på web)