

# INF1000: Grunnkurs i objektorientert programmering

Uke 0, høst 2014



## Oversikt

- Informatikk - hva & hvorfor?
- Datasystemer, maskinvare, programmer
- Objektorientert programmering i INF1000
- Programmeringsspråket Java
- Praktisk om INF1000
- Dette skal du kunne etter uke 0

## Utenom INF1000 Informasjon & hjelp

- Administrativt: Regler, påmelding, fravær, eksamen etc
  - <http://www.mn.uio.no/ifi/studier/admin/>
  - Ekspedisjonen i OJD 4. etasje
- Teknisk: Rom, utstyr, programvare etc
  - <http://termvakt.uio.no/OJD>
  - Resepsjon/ terminalvakter i OJD 1. etg
- Egen maskin: Laptop-hjelp
  - <http://termvakt.uio.no/Studielaben>
  - Resepsjonen OJD 1. etg, 18.8 - 12.9.14
- Obliger: Se semestersiden!

## Informatikk

*Informatikk er læren om hvordan datasystemer konstrueres og brukes\**

et **datasystem** består av en eller flere **datamaskiner** som kjører **programvare** og kan være knyttet til et eller flere **nettverk** for overføring av data.

IT: Informatikk handler om teknologi, men også mye mer!

\* Sitat dekan ved MatNat Morten Dæhlen

## Er informatikk viktig?



### Regjeringen oppnevner digitalt sårbarhetsutvalg

Et utvalg som skal kartlegge samfunnets digitale sårbarhet oppnevnes i statsråd fredag. Rapporten skal være klar i september neste år.

NTB

Oppdatert: 19. jun. 2014 07:58

Del Tweet E-post Legg arkivertjen i favoritter

- Det er snart 15 år siden forrige sårbarhetsutvalg la fram sin rapport. Siden da har det skjedd enormt mye som påvirker den digitale sårbarheten vår. Så å si alt i det moderne Norge er digitalisert. A nedsette et nytt utvalg nå er veldig viktig, slik at de får kartlagt hvor stor sårbarheten er og foreslått konkrete tiltak som kan redusere denne, sier justis- og beredskapsminister

Olav Lysne, Simula Research/ Uio

## .. fullt av muligheter?

Grøtt og blått  
IT i medisin og helse  
Multimedia  
Nanoelektronikk og robotikk  
Spillteknologi

Finne forskningsgrupper, forskningsprosjekter eller forskningsmetoder

Løser dataalderens tøffeste utfordring  
Oppnevnt utvalg skal gjøre det raskere og enklere å få viktig informasjon ut av enorme databaser

Saker fremtiden for Microsoft  
Fremtidens sikkerhetsoperatører blir smartere. Forskere ved KTD-laboratoriet hjelper Microsoft med å vinne kampen om sikkerheten

Musikk + IT = kreativ boom  
Valley som påvirker CP-komplekse barn, musikk som kan utløse prosedyrer, utvikling av helse med innvendige trommer. Foto: Gunnhild M. Haugen

Innovasjon med 3D  
Verdens største på helsedata  
Kjeldre Erna er professor ved helse og informasjon  
Definisjonen på innovasjon i forskning er ikke kommersialisering, men at noe tas i bruk. Innehar helse: Foto: Ole Sævi/USO

Innovasjon med 3D  
Fra Blindern til Silicon Valley  
Tobias Dahl var student ved Blindern for informasjon. I dag er han grunnlegger bak Elgitec Labs, an ledende med 20 ansatte i Oslo og Silicon Valley. Foto: Ole Sævi/USO

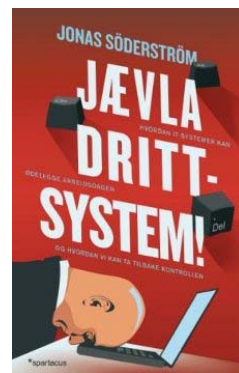
Aktuelle saker innen forskning

tifeldig utdrag fra Ifi's forside Forskning

## .. krevende?

Fra Stortingsmelding 10( 2012-2013):  
**God kvalitet – trygge tjenester**  
Kvalitet og pasientsikkerhet i helse- og omsorgstjenesten:

«... Eksempler på områder der Norge skårer dårlig er informasjonsflyt, kommunikasjon og koordinering mellom ulike deler av helsetjenesten...»



## Programmering

Hva handler det om?

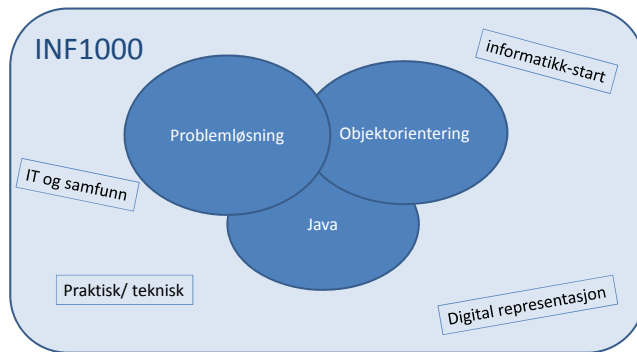
- formulere problemer
- tenke kreativt omkring løsninger
- og formulere løsninger klart og nøyaktig



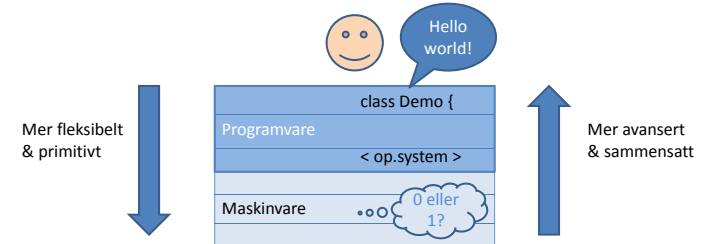
Programmering er problemløsning - og innebærer å leve mye av tiden med "problemer".

Eller: .. med "å skape noe nytt!"

## Grunnkurs i objektorientert programmering



## Lag på lag på lag...

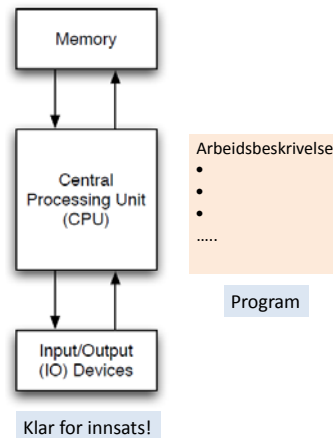
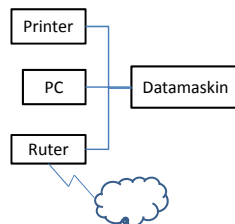


... mellom CPU'en og brukeren.



## Datasystem

en eller flere datamaskiner som kjører programvare og kan være knyttet til et eller flere nettverk (og som samarbeider om en oppgave)



## Programmering

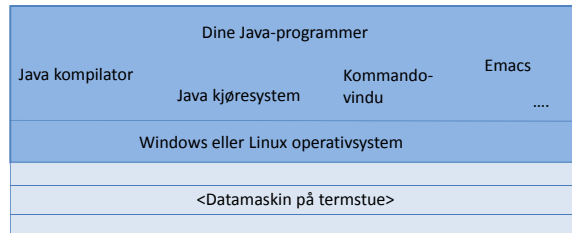
Å lære en datamaskin å gjøre noe nytt ved å gi den en detaljert oppskrift i et språk den forstår.

I INF1000 gjør vi det på en [Linux](#) eller [Windows-maskin](#), ved å

- å lage en fil med [Java](#)-instruksjoner f eks i [emacs](#)
- [kompilere](#) denne filen (oversette til maskinspråk)
- (rette feilene vi skrev inn i filen)

Programmet kan nå [kjøres](#) - da gjør maskinen det vi har bedt om!

## «Våre» lag



## Java - hva trenger vi?



- Kjøresystemet (Java Runtime Environment, JRE) trengs for å utføre programmene. Hvis man bare kjører ferdig kompilerte programmer er dette nok.
- Kompilatoren og annet som trengs av de som *skriver programmer* inngår i Java Development Kit (JDK) sammen med JRE og mye mer.

## Programmeringsspråket Java

- Mye brukt
- Persist/ sikkert - lar deg ikke gjøre «farlige» feil
- Samme programkode kan kjøre på flere maskiner
- Støtter sentrale konsepter for programmerere
  - Kraftfullt, men kan kjennes omstendelig
  - Undervisning: Nyttig, men krevende i starten
- NB: Java er ikke JavaScript

## Et første Java-program

```
class Uke00 {
    public static void main (String [] args) {

        // Dette programmet gjoer ingenting!

        /* Kommentarer kan vi ogsaa skrive
           på denne maaten, over flere linjer */
    }
}
```

## og ett til ...

```
class Uke01 {
    public static void main (String [] args) {

        // Skriver ut en tekst til brukerens skjerm:

        System.out.println ("Velkommen til INF1000!");

    }
}
```

## og ett til ...

```
class Uke03 {
    public static void main (String [] args) {

        // Konkaterer (skjoeter) to tekster
        // og skriver ut til skjerm:

        System.out.println ("Velkommen til" + "INF1000!");

    }
}
```

## og ett til ...

```
class Uke02 {
    public static void main (String [] args) {

        // Skriver ut samme tekst til brukerens skjerm:

        System.out.print ("Velkommen til ");
        System.out.println ("INF1000!");

    }
}
```

**Feil - uunngåelig & lærerikt**

- Fordi datamaskiner er maskiner er de lite tolerante og lite forståelsesfulle (om ikke de er programmert til å virke slik)
- Når vi programmerer vil vi heller ikke at maskinen skal begynne å «gjette» hva vi mener (mer akseptabelt ved googling!)
- Dvs strenge krav til nøyaktighet for at alt funker
- Kompileringsfeil (typisk skriveleifer)
- Kjøretidsfeil (noe går feil underveis)
  - Logiske feil (programmet gjør noe annet enn ventet)
- Tips foreløpig: UNNGÅ NORSKE TEGN HELT

# Linux

Anbefales

<log inn>

```
siriamj@eidsberg ~ $ cd Ifi/Programmering/2014
./Ifi/Programmering/2014
```

```
siriamj@eidsberg ~/Ifi/Programmering/2014 $ emacs Uke01.java &
[2] 5141
```

```
siriamj@eidsberg ~/Ifi/Programmering/2014 $ javac Uke01.java
```

```
siriamj@eidsberg ~/Ifi/Programmering/2014 $ java Uke01
Velkommen til INF1000!
```

```
siriamj@eidsberg ~/Ifi/Programmering/2014 $
```

## INF1000 semesterplan

| INF1000 | 2014 | Mål for uka  | Forelesing  | Obl |
|---------|------|--|---|-----|
| Uke 0   | 34   | Komme i gang - skrive, compilere, kjøre miniprogram  | System.out.print/ println<br>Variable, keyboard input,<br>forgreining |     |
| Uke 1   | 35   | Programmering er problemløsning!   |   |     |
| Uke 2   | 36   | Kontrollflyt, feilsøking   | Feilmeldinger. Void-metoder   | 1   |
| Uke 3   | 37   | Hvordan løse problemer med programmering?  | Enkel looping og enkle arrays   | 2   |
| Uke 4   | 38   | Hvordan løse problemer som involverer data?  | Filer og tekst. Value methods   | 3   |
| Uke 5   | 39   | Hva er objekt-orientert programmering?   | Klasser, objekter og pekere   | 4   |
| Uke 6   | 40   | Hvilken rolle spiller IT i samfunnet?  |   | 5   |
| Uke 7   | 41   | Hvordan lage objektorienterte programmer?  | Instans-metoder.<br>Mer om skop, typer,<br>konvertering.              | 6   |
| Uke 8   | 42   | Hvordan lage mer komplekse programmer?   |   |     |
| Uke 9   | 43   | Hvordan kode for skalerering til store systemer?<br>Hvordan representeres tall, tekst, bilder og mer på en<br>datamaskin? Hvordan håndtere personvern? |   | 7   |
| Uke 10  | 44   |  |   |     |
| Uke 11  | 45   | Et større program  |   |     |
| Uke 12  | 46   | Repetisjon   |   |     |
|         | 47   | Prøveeksamen   |   |     |
|         | 48   |  |   |     |
|         | 49   | Eksamen..  |   |     |

# Windows

Funker også

- finn emacs på Start-menyen (under Gnu Emacs), start dette
- <les inn fil eller skriv nytt program, lagre når ferdig>

- finn Command Prompt på Startmenyen (under Tilbehør)
- gå til riktig disk, f eks M: `C:\Users\siriamj> M:`
- gå til riktig mappe med CD-kommandoen

```
M:\>cd Ifi\Programmering\2014
```

- Kompiler programmet ditt med javac

```
M:\Ifi\Programmering\2014> javac MittProg.java
```

- Utfør programmet

```
C:\> java MittProg
```

## «Normal» undervisningsuke

|       | Mandag                             | Tirsdag | Onsdag                  | Torsdag                        | Fredag | Helg |
|-------|------------------------------------|---------|-------------------------|--------------------------------|--------|------|
| 08:15 |                                    |         |                         |                                |        |      |
| 09:15 |                                    |         |                         |                                |        |      |
| 10:15 |                                    |         |                         |                                |        |      |
| 11:15 |                                    |         |                         |                                |        |      |
| 12:15 | INF1000-<br>gruppe.<br>2 t seminar |         |                         | INF1000-<br>gruppe.<br>2 t lab |        |      |
| 13:15 |                                    |         |                         |                                |        |      |
| 14:15 |                                    |         | INF 1000<br>forelesning |                                |        |      |
| 15:15 |                                    |         |                         |                                |        |      |
| 16:15 |                                    |         |                         |                                |        |      |
| 17:15 |                                    |         |                         |                                |        |      |

Forventet arbeidsinnsats:  
13-14 timer/ uke/ emne



PROGRAMMERING  
- Oblig. oppgaver  
- Øvingsoppgaver

## «Uke 0»

Kalender-  
uke 34

| Mandag1<br>8.8 | Tirsdag<br>19.8 | Onsdag<br>20.8                     | Torsdag<br>21.8                           | Fredag<br>22.8 | Helg |
|----------------|-----------------|------------------------------------|---|----------------|------|
|                |                 | INF1000<br>Fore-<br>lesning<br>2 t | Frivillig lab<br>«Forkurs-trening»<br>2 t |                |      |

Kalender-  
uke 35

| Mandag<br>25.8      | Tirsdag<br>26.8  | Onsdag<br>27.8                     | Torsdag<br>28.8               | Fredag<br>29.8 | Helg |
|---------------------|------------------|------------------------------------|-------------------------------|----------------|------|
| Øvings-<br>oppgaver | Lærebok<br>uke 1 | INF1000<br>Fore-<br>lesning<br>2 t | INF1000-<br>gruppe<br>2 t lab |                |      |

## Neste uke

- Programmering som problemløsning
  - variable, forgreninger og innlesing fra bruker



Lykke til med semesteret og programmeringen!

## Hva bør du kunne etter uke 0?

- Følg med på [semestersiden](#) for
  - Praktisk informasjon og beskjeder
  - Krav til obliger, innlevering
  - Undervisningsplan
    - Pensum og ressurser
    - Flervalgstester for teori
    - Programmeringsoppgaver for praksis, inkl obliger
- [Flervalgstest for uke 0](#)