

# IN1000: Introduksjon til objektorientert programmering

Uke 1

Introduksjon til faget og emnet

# I dag

- Informatikk og noen nyttige begreper
- Praktisk informasjon og tips for IN1000
- <Pause>
- Første innføring i programmering

*Mye programmeringserfaring? Se slide 18 😊*

# Hvem er vi?

Gruppelærere:  
20 stk inkl  
sjefgruppelærere  
Marlen  
Peter

Ekstra ressurser:  
Lonneke  
Ivar

studenter

Rettere:  
8 stk

Faglærere:  
Geir Kjetil (høst)  
Siri (høst)  
Stein Michael  
Henrik

# Dine medstudenter

(fra Forkurs for IN1000, 2020)

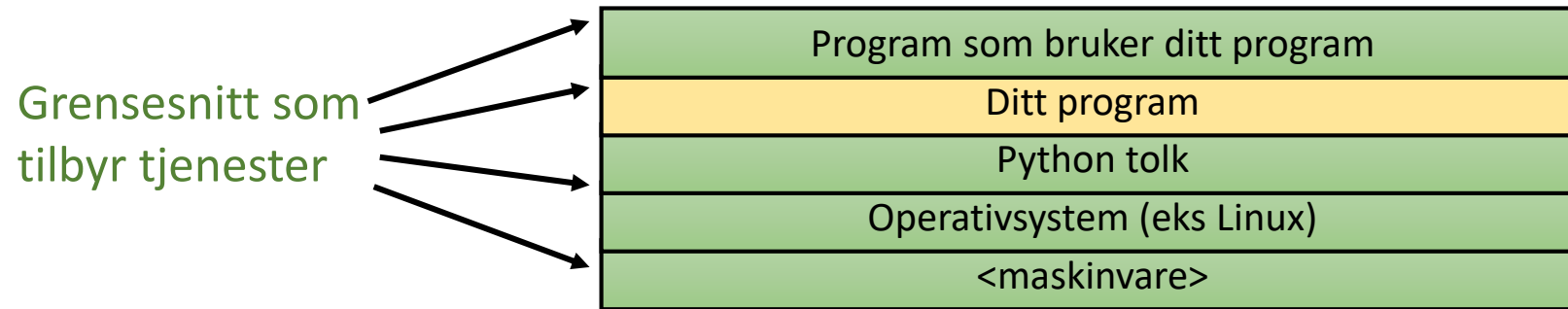


# Informatikk

***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**

# Din maskin: Lagdeling og grensesnitt



- Hvert **lag** har sine arbeidsoppgaver, og tilbyr disse gjennom et **grensesnitt** til laget over
- Grensesnittet gjør det mulig å **abstrahere** bort (ikke kjenne til) *hvordan* laget under løser oppgavene sine

# Abstrahere/ abstraksjon

## Wikipedia:

- "..tenkemåter der en skiller ut enkelte elementer, for å betrakte resten.."
- "..Det er en måte å komme fram til fellestrekk eller felles egenskaper og klassifiseringer ..."

## Store Norske Leksikon:

- "å sammenfatte enkeltfenomener under ett begrep idet man ser bort fra avvikende detaljer"

# Modell

SNL:

<...> brukes ordet om en skjemativering eller en matematisk modell

Slik brukt innebærer en modell alltid en **forenkling** basert på et utvalg av de viktigste trekk.

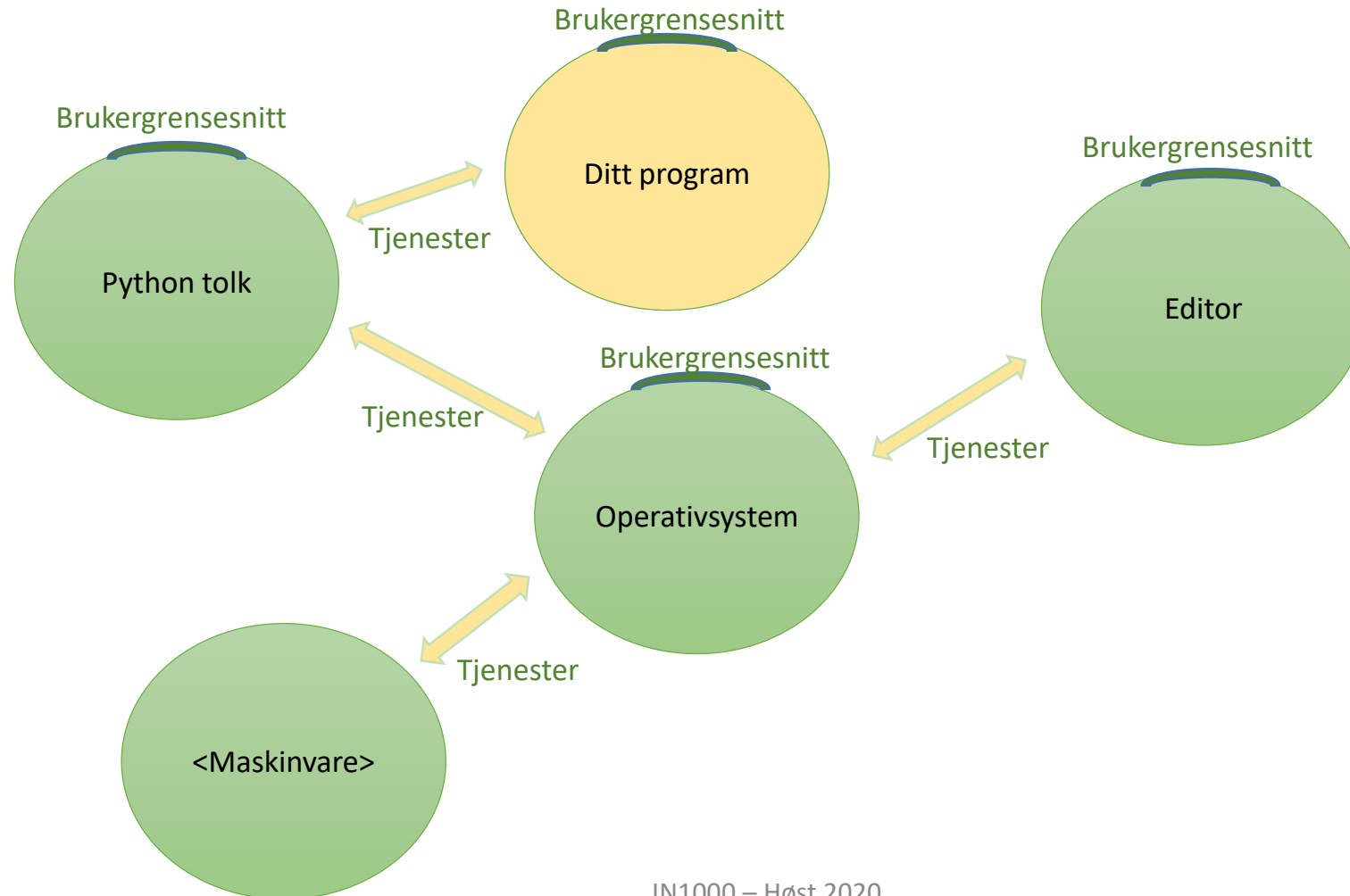
Wikipedia:

**Teoretiske modeller** gir et **sterkt forenklet** bilde av virkeligheten.

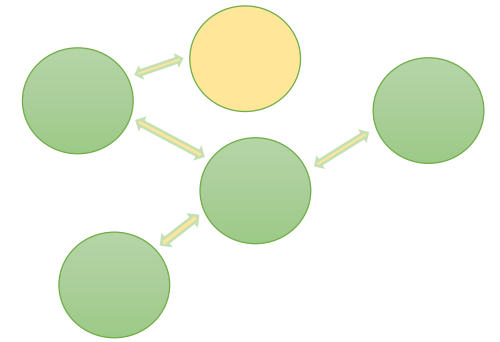
Forenklingen dreier seg om å fokusere på de **trekk som anses å være vesentlig** for det som skal forklares, samtidig utelukke alt som er uvesentlig



# En annen modell av hvordan deler i et system samarbeider



# Objektorientert programmering



- Når programmet ditt utføres, foregår det (bak kulissene) vha **objekter** som samarbeider.
- Et objekt holder rede på **data** som hører sammen, og kan utføre **handlinger** knyttet til disse
- *Hvilke* data og handlinger er definert i objektets **klassedefinisjon**
- Objektene tilbyr **tjenester** via **metoder** til andre deler av programmet gjennom sine **grensesnitt**
- Objektorientering hjelper oss å **organisere** programmene våre bedre, ved å **modellere** og **simulere** komplekse systemer

**Vi kommer tilbake til hvordan vi kan opprette og bruke objekter, og hvordan vi kan skrive egne klasser som mønstre for objekter!**

# Oppsummering

- Informatikk *krever* at vi abstraherer bort forskjeller og kompleksitet vi ikke trenger i det vi jobber med "akkurat nå"
- For eksempel vha modeller og metaforer som beskriver hva som skjer "bak kulissene" når programmene våre utføres
- Et fenomen kan modelleres på ulike måter avhengig av hvilke egenskaper/aspekter man er interessert i (trenger å vite noe om)
- **Objektorienterte språk** har mekanismer som hjelper oss med dette når vi skriver programmer

# Informatikk-studiet og IN1000

- Mål for emnet
  - Solid grunnlag for videre studier
  - Vekt på grunnleggende konsepter og begreper
  - Trening i programmering ved hjelp av disse i Python
  - Ikke et hurtigkurs i praktisk programmering – eller i Python
- Ingen krav til forkunnskaper, men helt nødvendig med jevnt arbeid!
  - Gjenspeiles i de obligatoriske oppgavene som kreves i emnet

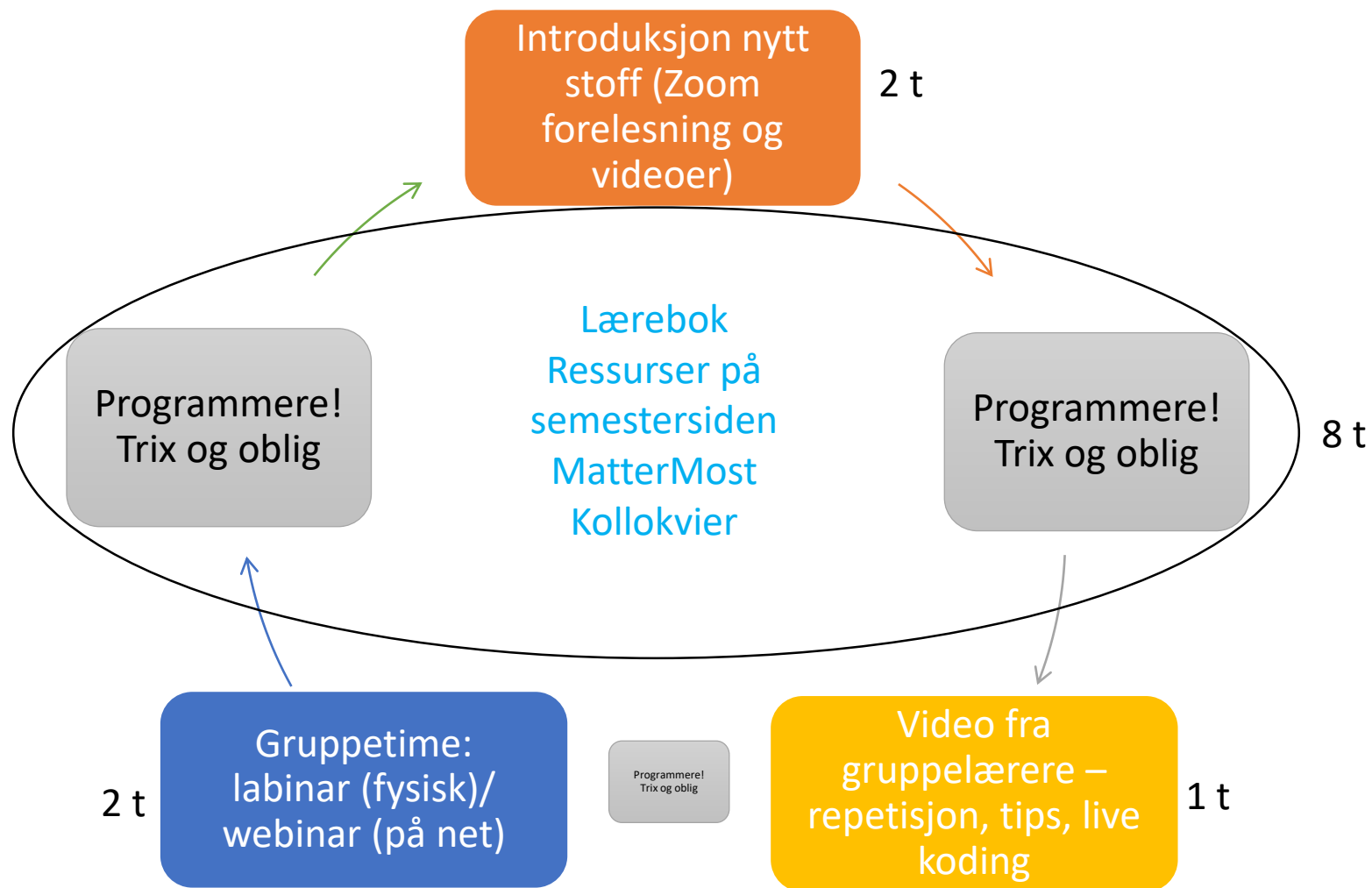
# Semesteret i IN1000

- Uke 1-6: Grunnleggende elementer i programmering
- Uke 7-12: Objektorientert programmering, større programmer, mer kompleksitet
- Uke 13-15: Prøveeksamen, eksamenstips, repetisjon
- Uke 16: Eksamen

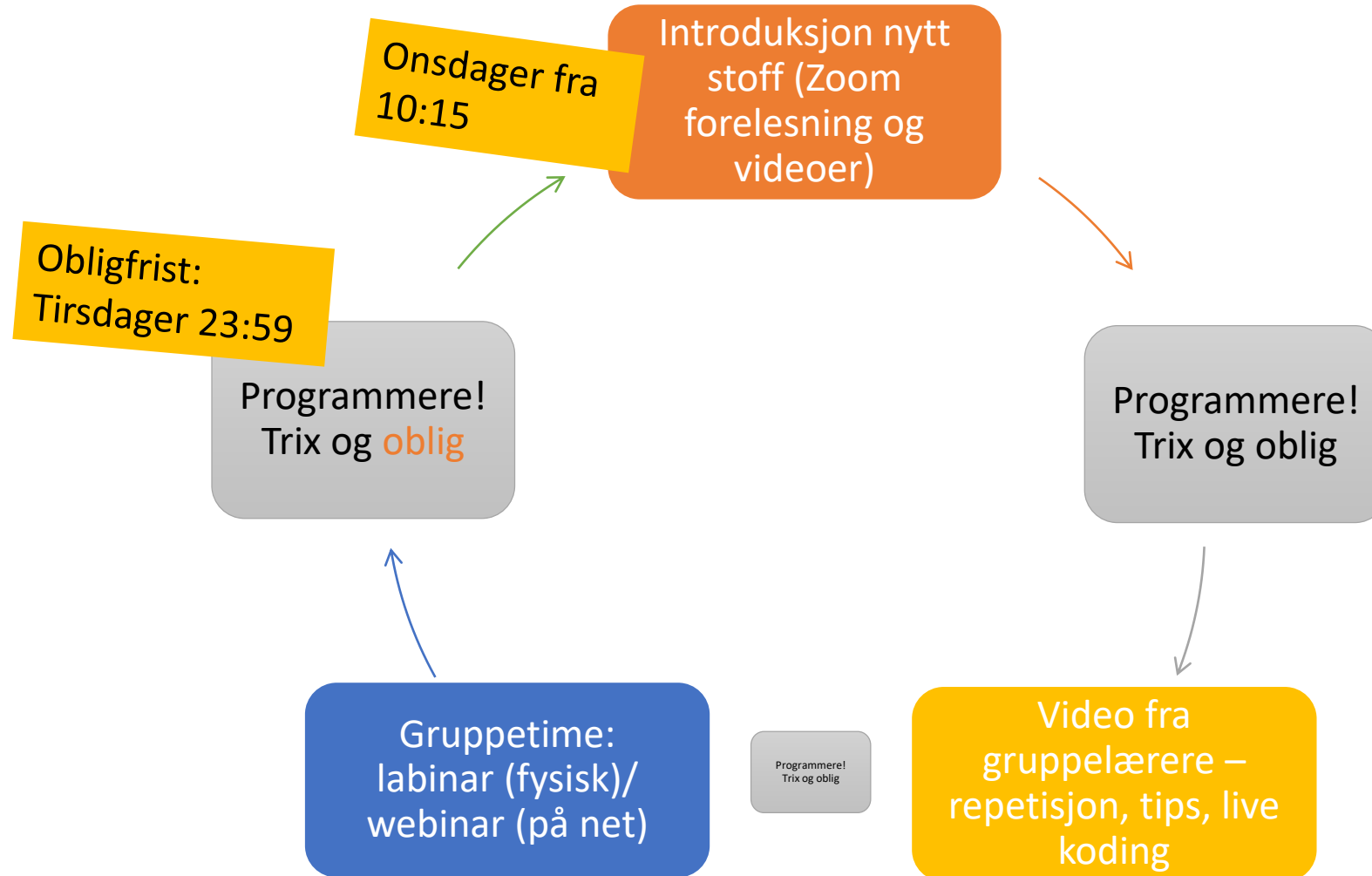
# Obligatoriske innleveringer

- Alt leveres i Devilry.ifi.uio.no (demo i gruppetime)
- Harde frister, stenges ved fristen
- Alle frister tirsdag kl 23:59
- Lever gjerne flere ganger, siste gjelder
  
- 1-6: ukes-obliger, disse gir poeng
  - Samlet for alle kreves ca 2/3 av max poengsum. 100% er faglig minimum!
  - Dette er en «reserve» i tilfelle sykdom, teknisk krøll etc
  - Ingen utsettelse fra gruppelærer (sykdom e.l: studieinfo@ifi.uio.no)
  - Ingen nye forsøk
  
- 7-8: større obliger, begge må godkjennes
  
- NB: Krav til selvstendig arbeid, som kan forklares

# Arbeidet i IN1000 – ca 13 t/ uke



# Arbeidet i IN1000 – ca 13 t/ uke





# Tips for programmeringsstudier

- Programmering handler om å definere og løse problemer/ finne fremgangsmåter, dvs
  - Læringen skjer mens du klør deg i hodet
  - Lær å like det 😊
- Start koding selv om du ikke forstår alt, eksperimenter
- Men finn alltid ut hvorfor en feil "forsvant"!
- Jobb sammen med andre, diskuter løsninger og "hva skjer her"
- Jobb alene, svett over problemer 😊
- Du kan ikke hoppe over pensum i programmering

# Forkunnskaper eller lyst på mer utfordringer?

- Noen fallgruver
  - Følg undervisning og løs obliger, ikke «mist toget»
- IN1000-gruppe med alternative oppgaver
  - Starter opp uke 3, følg med på IN1000 semestersiden
- Mye bakgrunn i Java? Infomøte/ opptaksprøve
  - IN2030 - Prosjektoppgave i programmering
  - Kun førsteårs-studenter på Ifi programmer
  - Fredag 21.8 kl 13:15 på Postscript, ta med egen laptop (m/ JDK)
  - Spørsmål til Dag Langmyhr ([dag@ifi.uio.no](mailto:dag@ifi.uio.no))
- Overskudd til ekstra studiepoeng?
  - IN1150 – Logiske metoder (nettundervisning høstsemesteret)

