



Forelesning 1

INF101 - 17/1 2022





Innhold

- Hvem underviser INF101?
- Hva er programmering?
- Hva er god kode?
- Praktisk
 - Hva forventes av studentene?
 - Hva kan dere forvente av oss?
- Objekt orientert programmering
- Java vs Python
 - Variabler i Java





Foreleser

- Martin Vatshelle
- Doktorgrad i Algoritmer ved UiB
- Har bodd 2 år i Canada
- Har jobbet 5 år i Oljebransjen
- Startet å jobbe for UiB høsten 2019





Foreleser

- Torstein Strømme
- Fagbrev i elektronikk
- Studerte 4 år i USA etter VGS
- Doktorgrad i algoritmer ved UiB
- Jobbet med IT og programmering
 - 2 år i oljebransjen
 - 1 år i media-bransjen.
- Startet å jobbe for UiB høsten 2021





Undervisningsassistenter

- Laura Garrison
 - Administrativ hjelp
- Sondre Bolland
 - Oppgavegjennomgang
 - Undervisningsopplegg
 - Får hjelp av Sophie Bosio
- Eric Mörth
 - Oppfølging av gruppeledere og studenter





Gruppeledere

- Daniel Berge
- Dusan Nikolic
- Elias Bendixsen
- Endre Sletnes
- Hadi Alkadiri
- Hannah Mørken
- Jakob Snorrason
- Jon Vegard Lokøy
- Jørgen Storum
- Magnus Tønnesen
- Magnus Vestvik
- Martin Andvik Øvsttun
- Maya Robbestad
- Sophie Bosio
- Stian Munkejord
- Sven Alrik Solemdal
- Tellev Sundt
- Thomas Thunes
- Tobias Madsen
- Østen Edvardsen





Hva er programmering

- Programmering er å få en PC til å gjøre en oppgave.
- Vi tenker ikke likt som en PC jobber.
 - PC trenger nøyaktige instruksjoner
- Programmeringsspråk er designet for å gjøre det lettere å instruere PCen





Hvorfor trenger vi programmeringsspråk?

Maskin kode:

```
0x 60 00 00 80
0x A4 00 00 00
0x 60 01 00 84
0x A4 01 01 00
0x 60 02 00 00
0x 60 03 00 04
0x 60 04 00 00
0x 60 05 00 01
0x 08 00 00 02
0x 20 00 00 03
0x 20 04 04 05
0x 11 20 04 01
```

=

```
01100000000000000000000010000000
10100100000000000000000000000000
...
```



- Ikke enkelt å programmere i maskin kode!
- Hver prosessor har forskjellig maskin kode!



Hvorfor trenger vi programmeringsspråk?

- Assembler kode:

```
IMM R0, 0x80
LOAD R0, R0
IMM R1, 0x84
LOAD R1, R1
IMM R2, 0x0
IMM R3, 0x4
IMM R4, 0x0
IMM R5, 0x1
STORE R0, R2
ADD R0, R0, R3
ADD R4, R4, R5
BNE 0x20, R4, R1
```





Hvorfor trenger vi programmeringsspråk?

Byte kode:

```
.method public static main
  ([Ljava/lang/String;)V
.limit stack 4
.limit locals 3
.line 3
  iconst_3
  anewarray java/lang/String
  dup
  iconst_0
  ldc "Viva"
  astore
  dup
  iconst_1
  ldc "Las"
  astore
  dup
  iconst_2
  ldc "Vegas"
  astore
  astore_1
.line 4
  iconst_0
  istore_2
  goto 139
.line 5
125:
  new Viva
  dup
  aload_1
  iload_2
  aaload
  invokespecial Viva/<init>
    ([Ljava/lang/String;)V
  pop
.line 4
  iinc 2 1
139:
  iload_2
  aload_1
  arraylength
  if_icmplt 125
.line 7
  return
```

Fordelen med «Byte kode» er at den er lik for alle maskiner.





Java kode

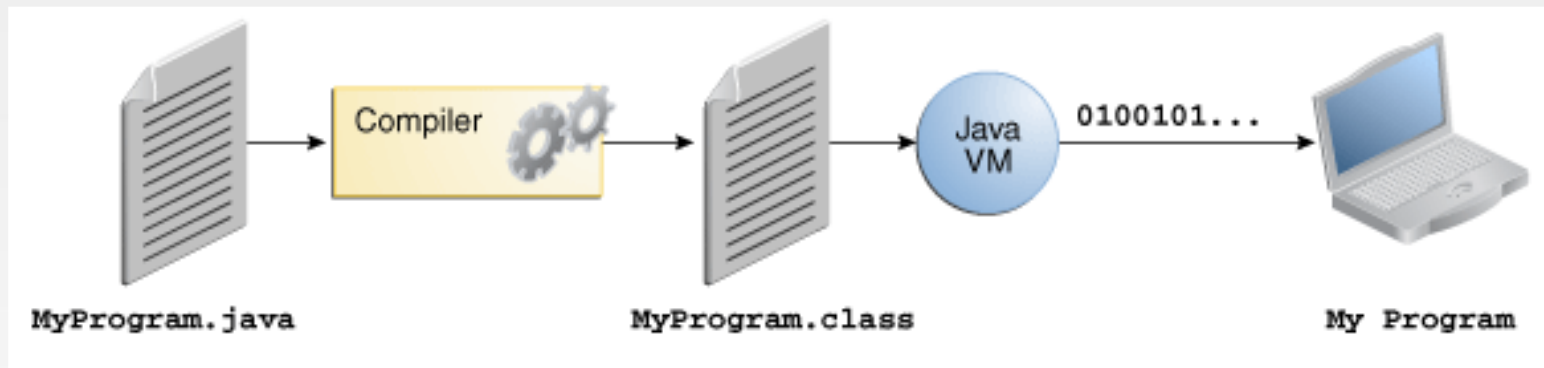
```
if(a > 5)
    a = a+3;
else
    a = a-3;
```

- Java kode kan se slik ut.
- Det er det vi skriver
- Programmet javac.exe omformer Java kode til Byte kode





Java fra kode til program





Installere Java og IDE

- Lab 0 handler om installering
 - Ligger ute på Mitt UiB
- Valg av IDE
 - Visual Studio Code
 - Eclipse
 - IntelliJ
- Java (Open JDK)





Komme i gang med Java

- Mye er likt som Python
- Alt dere har lært om Python skal dere også kunne om Java
- Basic Java kurs denne uken med Sondre Både Tirsdag og Onsdag
- Websiden codingbat.com/java er et fint sted å begynne for å komme i gang med Java.





Hva er god kode

- Funksjonell
- Feilfri
 - Hvorfor oppstår bugs?
- Rask (dette lærer dere i INF102)
- Lett å utvide
 - Lett å lese/forstå
 - Gjenbruk av kode
 - Modulær





Hvorfor oppstår bugs

- Misforståelse angående hva et program skal gjøre
 - Dårlig dokumentasjon
 - Ikke lest dokumentasjon
 - For komplisert til å forstå hva en skal gjøre
 - Spesialtilfelle som ikke er tenkt på
 - Dårlig forståelse av standard Java
 - Duplikat kode





Timeplan INF101

- Mandag 12:15-14 – Forelesning
- Tirsdag 12:15-14 – Forelesning
- Ukesoppgaver gis ut nesten hver uke
- Dere jobber på egen hånd og går på gruppetime for å få hjelp.
- Onsdag 16:15-18 - Oppgavegjennomgang
- Torsdag 13-14 – Spør foreleser
- Mandag og Tirsdag – Oppfølging avtales med de studentene det er aktuelt for.





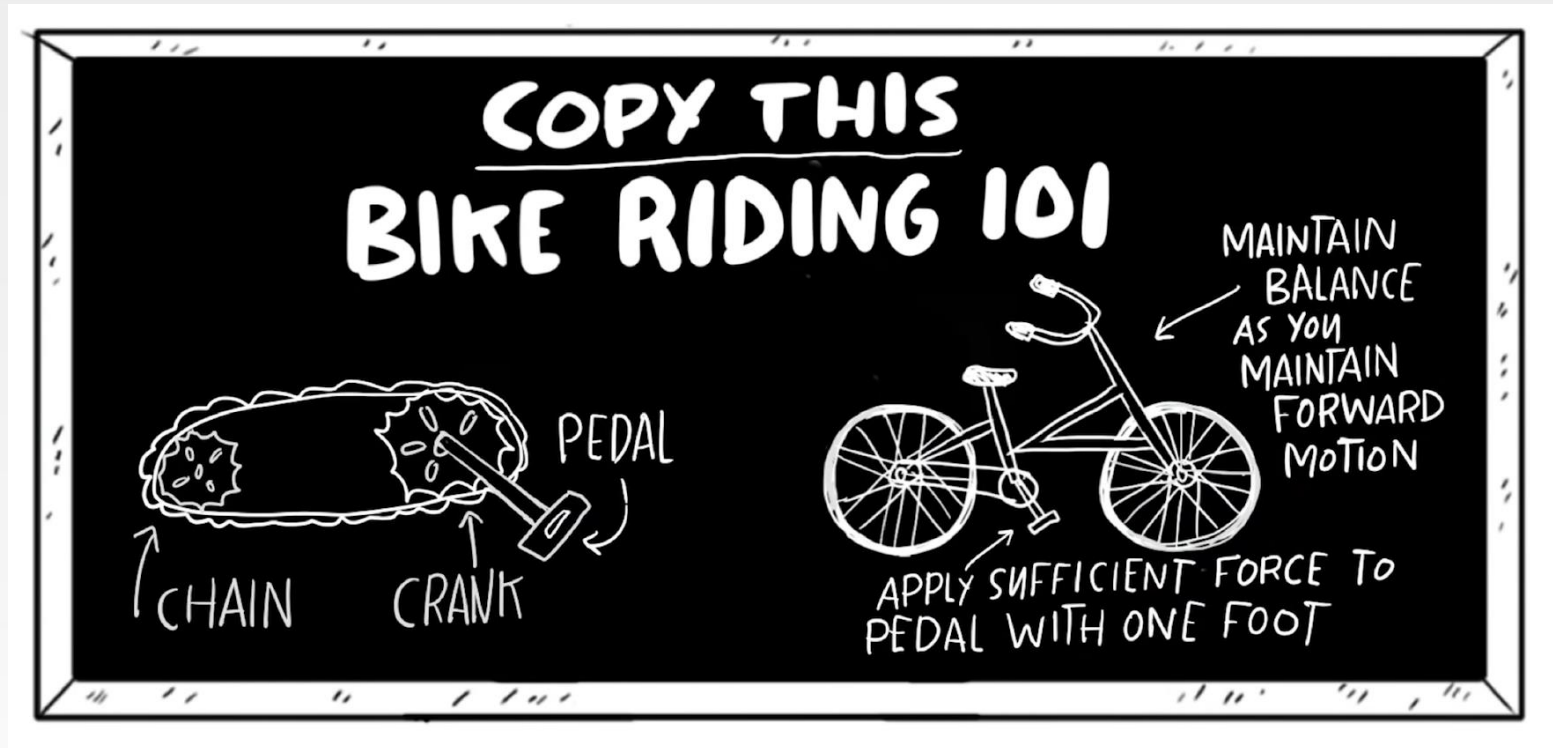
Ukentlige øvinger

- Nesten hver uke blir det gitt øvingsoppgaver
 - Dere bør begynne på oppgavene på egen hånd
 - Gå på gruppeøvingene!
 - Gruppeøvingene starter med repetisjon av uken der dere får stille spørsmål.
 - Så blir det individuell hjelp til de som ikke har fått til ukesoppgavene.
-
- Uke-oppgavene er obligatoriske.
De dekker pensum godt så det lønner seg å gjøre!





Ikke alt bør læres på forelesning



Bilde lånt fra en [Tedx talk](#)





Stille spørsmål

- Discord server
 - Informasjon om din gruppe
 - Stille spørsmål til gruppelederne
 - Stille spørsmål til medstudenter
 - Sondre Bolland administrerer





Hvis gruppeleder ikke kan svare

- Gruppeledere kan ikke svare på alt
- Når de står fast skal de ta spørsmålet videre til andre gruppeledere eller foreleser
- Foreleser vil av og til svare på diskusjoner og spørsmål på discord
- Gruppeledere er bare betalt for et par timer i uken så dere kan ikke forvente at gruppeledere er tilgjengelig på discord 24/7





Obligatoriske øvinger

- Det blir gitt 2 Obligatoriske øvinger I kurset
 - Planlagt dato for utlevering er:
28. Februar og 28. Mars
- Hver av de obligatoriske øvingene teller 15% av total karakter
- Obligatoriske øvinger må være bestått for å få ta eksamen





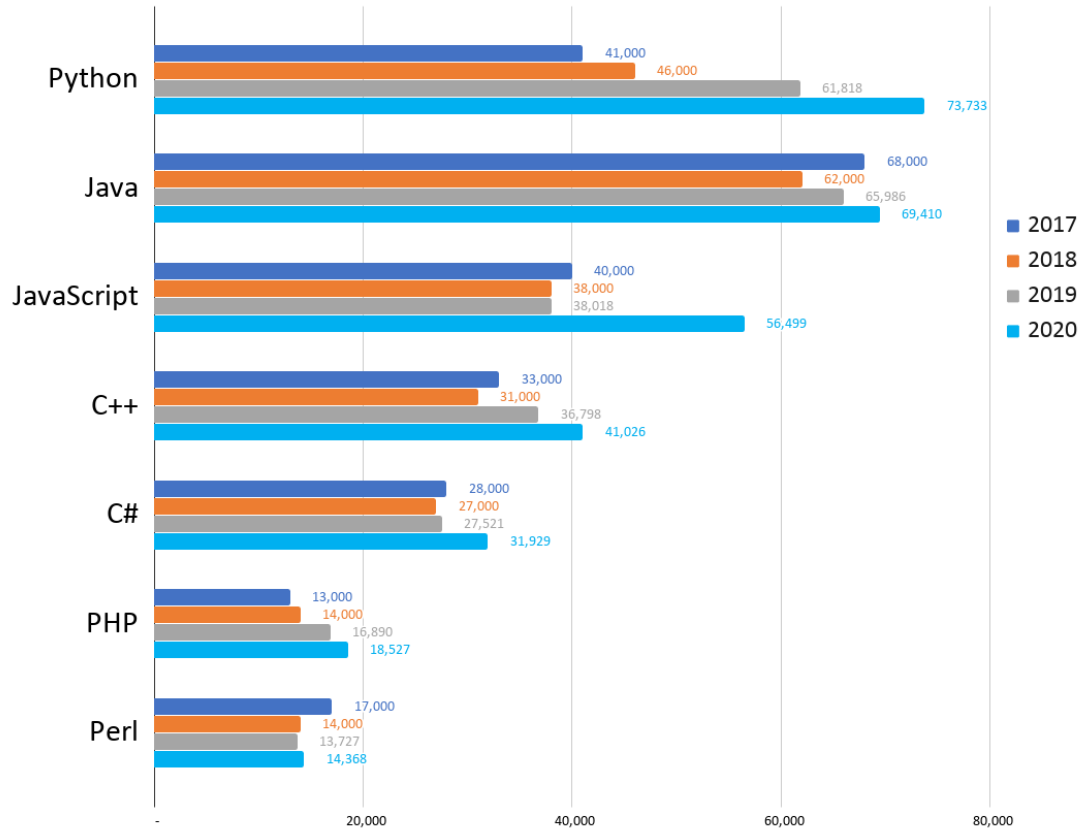
Oppgavegjennomgang

- Onsdag 16:15 – 18:00
- Emner kan være f.eks:
 - Gjennomgang av ukeoppgaver
 - Triks og tips i bruk av editor
 - Repetisjon av emner som studenter ønsker.





How do our usual languages fare?
Worldwide jobs on indeed.com





Hva er typer?

- Alle variabler har type (både i Python og Java)
 - For eksempel String og int
- Python kan lagre hva som helst i en variabel
 - «Dynamically typed»
- Java kan kun lagre den typen som variabelen er ment for
 - «Statically typed»





Java typer

- Python finner automatisk ut hvilken type en variabel er.
- I Java må vi bestemme typen når vi lager variabelen.
- Det er litt mer jobb i Java, men så har man også mer kontroll på hvilke typer variable man jobber med.





Java primitive typer

Primitive Type	Size	Minimum Value	Maximum Value	Wrapper Type
char	16-bit	Unicode 0	Unicode $2^{16}-1$	Character
byte	8-bit	-128	+127	Byte
short	16-bit	-2^{15} (-32,768)	$+2^{15}-1$ (32,767)	Short
int	32-bit	-2^{31} (-2,147,483,648)	$+2^{31}-1$ (2,147,483,647)	Integer
long	64-bit	-2^{63} (-9,223,372,036,854,775,808)	$+2^{63}-1$ (9,223,372,036,854,775,807)	Long
float	32-bit	32-bit IEEE 754 floating-point numbers		Float
double	64-bit	64-bit IEEE 754 floating-point numbers		Double
boolean	1-bit	true or false		Boolean
void	-----	-----	-----	Void





Java eksempel

- La oss lage noen program I Eclipse





Divisjon

```
int c = a/b;
```

- I uttrykket a/b
hvis både a og b er av typen `int`
så vil Java bruke heltallsdivisjon
D.v.s. runder ned til nærmeste heltall.
- $a /= 5;$
er det same som:
 $a = a/5;$





Er det flere operatører i Java?

- Når du lurer på noe spør



API

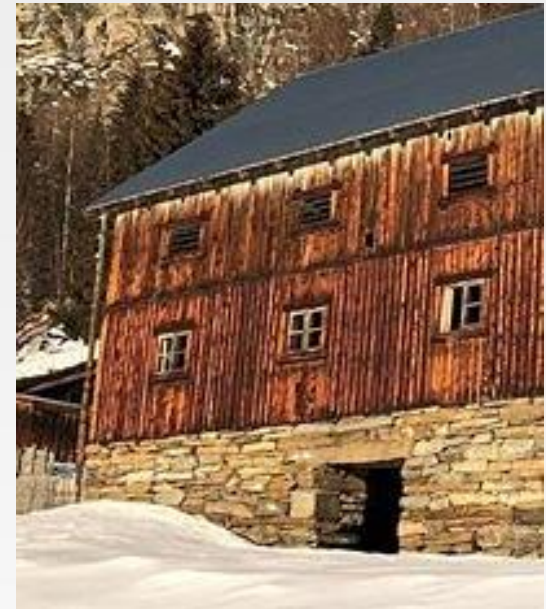


Google





Hva er viktigst å lære?



- Ta deg tid til å lære det grunnleggende ordentlig så du ikke får problemer senere