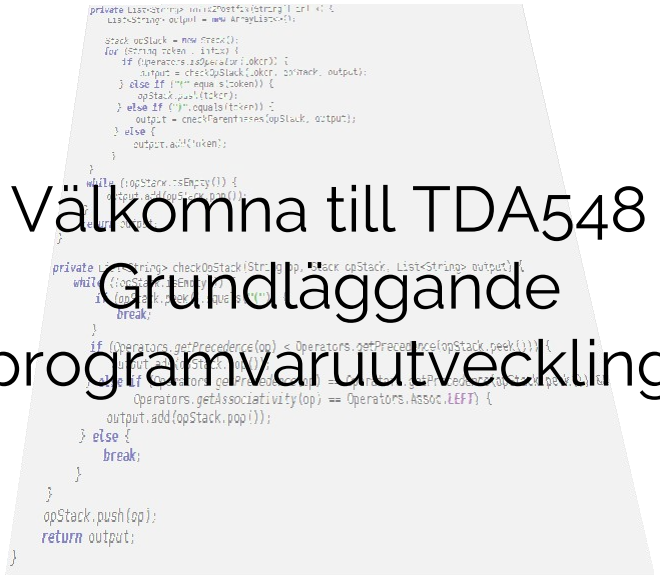


Välkomna till TDA548 Grundläggande programvaruutveckling

Lärare och Handledare

Kursansvariga, examinatorer, föreläsare och handledare

- **Joachim von Hacht**, hajo@chalmers.se, 772 1003

Handledare

- Joel Hultin
- Erik Norlander
- Sabrina Samuelsson
- Samuel Moos

Syfte med Kursen

Efter avslutad kurs skall studenten, givet ett problem, kunna implementera ett program som löser problemet.

Detta innebär att studenten:

- Kan tolka innebörden av och lösningen till det givna problemet.
- Kan tillämpa funktionell abstraktion och funktionell nedbrytning för att strukturera problemet.
- Kan, med hjälp av ett programmeringsspråk, implementera programmet på ett strukturerat, rimligt effektivt och begriplig sätt (idealt självdokumenterande).
- Kan använda en teststrategi.

Efter kursen skall studenten ha fått en grundläggande begreppsapparat för området.

Kommunikation

Allt ni behöver finns på kurssidan

- http://www.cse.chalmers.se/edu/course/tda548

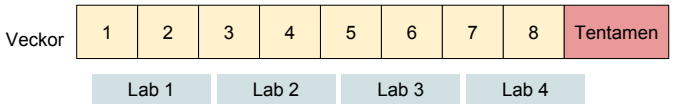
Meddelanden och nyheter läggs in efter hand

- Viktigt att besök sidan emellanåt
- Kursen går för första gången i denna form, ev. fel och oklarheter kan dyka upp, meddelas på kurssida

Det går, i alla sammanhang, alltid bra att fråga!

Planering

(4h föreläsningar, 4h övningar, 6-8h laborationer)/vecka



Lab 1 | Lab 2 | Lab 3 | Lab 4

Övningar och laborationer i datorsalar.
Grupper: A, B, C, D, se schema

5

Tyvärr är schemat ganska komplicerat. Kontakt mig ifall konstigheter dyker upp.

Övningar och Laborationer

Övningar

- Övningar förbereder för laborationer (en övningsbunt (zip-fil)/vecka)
- Behöver inte redovisas men diskutera gärna lösningar med handledare

Laborationerna

- Är obligatoriska.
- Ni redovisar laborationer för handledare under laborationspassen.
- Skall göras i grupper om 3 studenter.
- Ni godkänns som 3-grupp
- Alla i gruppen måste bli godkända (kunna svara på frågor) för att gruppen skall bli godkänd.

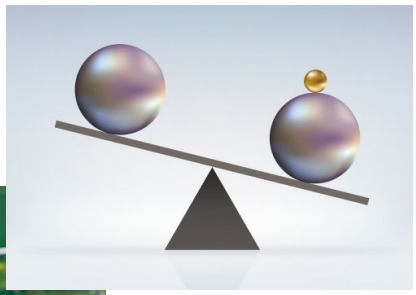
Lägg inte ut lösningar till övningar på nätet!

- Ni förstör för era kamrater!

Lägg inte ut lösningar till laborationer på nätet!

- Era kamrater riskerar att bli underkända ifall de använder utlagd kod!

Bra Labgrupper



7

I programmering förstärks skillnader i kunskaper.

- En liten skillnad mellan två personer kan göra att den ena får (mycket) svårt att hinner med.
    - Ungefär som badminton (tror jag)
- Att som nybörjare jobba ihop med en erfaren programmerare hjälper inte, tvärtom, det stjälper!
- Försök hitta kamrater som ligger på så samma nivå som möjligt.
- Som sagt: Ni laborerar och redovisar i 3 grupper.

Tentamen

Skriftlig tentamen

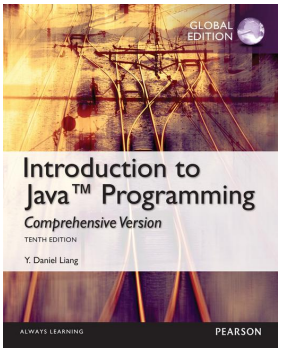
- Papper, penna och suddgummi
- Inga hjälpmedel (förutom lexikon engelska - valfritt språk)
- Om ni gjort, och förstått, övningarna och laborationerna skall det inte behöva tentamenpluggas … istället …
- … repetera övningar och laborationer

Examination

För att klara kursen måste du få alla laborationer godkända samt klara tentamen. Kursbetyg ges av tentamen enligt

Betyg	Poäng/Maxpoäng
5	0,8
4	0,63
3	0,4
U	< 0,4

Bok



10

Valfri bok, i bilden en engelsk som är bra, vill man ha en svensk tar man den andra

- Ni får själva matcha kapitel i böcker mot mina föreläsningsanteckningar.

Utvecklingsmiljö

Vi använder Windows och Linux, Java 8 och IntelliJ

- I skolan finns allt

Vill du ha miljön på egen dator?

- Java och IntelliJ finns för all operativ.
- Installera Java först
- Därefter IntelliJ

Java 8 = Java 1.8

Språk

Ord/uttryck	Typ av ord	Engelskt ord som åsyftas	Förslag på översättning
A			
attacha	v.	attach	bifoga
attachment	s.	attachment	bilaga
B			
backslash	s.	direkt inlån	bakåtvänt snedstreck, bakåtstreck, bakstreck [7]
backup	s.	direkt inlån	säkerhetskopia
body (i brev)	s.	direkt inlån	brödtext, (brev)innehåll, brev, brevkropp
browser	s.	direkt inlån	(webb)läsare
bug/bugg	s.	direkt inlån	fel, lus (om lus används behålls namnets historiska ursprung)
C			
cache	s.	direkt inlån	buffert(minne), mellanlager, mellanminne [9]
cancel/cancla	v.	cancel	avbryta
cc:a	v.	direkt inlån	skicka kopia (till)
chatta	v.	to chat	prata, tala, småprata, snacka
Connecta	v.	Connect/Connect to	Koppla upp, ansluta
D			
debugga	v.	to debug	avlusa, felsöka
delete:a	v.	to delete	ta bort, radera

Svårt område:

- Jag talar/använder det som känns bäst men försöker använda svenska (speciellt när det svenska ordet är “självförklarande”, enklare eller kortare).
    - Anger ofta begrepp på engelska också
- All kod (verkligen all) skrivs dock på engelska
- Se länk kurssida om olika svengelska uttryck.

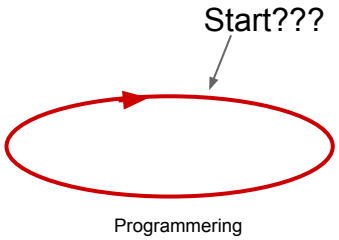
Att lära sig programmera 1

13

Programmering är inledningsvis (denna kurs) ett övningsämne.

- Jämför tennis eller windsurfing. Övning ger färdighet!
- Ju fler program du skriver desto bättre lär du dig
- Visst kan man (bör man) läsa i böcker, men det viktiga är att skriva program! Många!

Att lära sig programmera 2

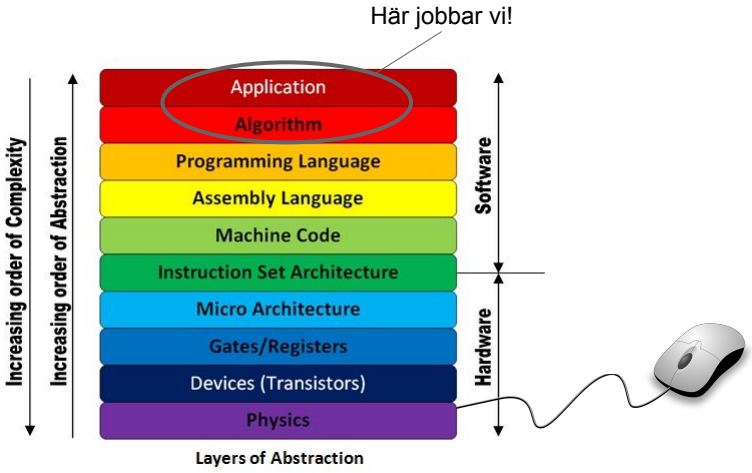


14

I t.ex. matematik finns en ganska tydlig ordning, vad som kommer först.

- I programmering finns ingen lika tydlig sådan, det finns alltid något före …
- Man får acceptera att man inte för tillfället förstår alla detaljer.
    - När kursen är färdig skall allt (som gäller denna kurs) var uppenbarat.

Att lära sig programmera 3



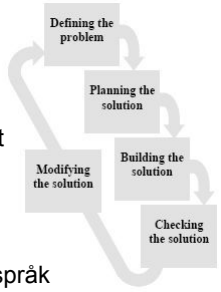
15

En dator jobbar på många olika abstraktionsnivåer.

- Vi kan omöjligt hålla alla dessa i huvudet samtidigt
- Man får acceptera att man inte kan förstå allt "i botten" just nu.
- Gör vi något i programmet så händer något, exakt hur detta sker på underliggande nivåer funderar vi inte på.
    - Det som sker på "vår" nivå skall vi ha stenkoll på!

Att lära sig programmera 4

Arbetssätt



Problemlösning

"The boss wants me to create a computer algorithm that converts hindsight into foresight."

Fackspråk

Kvalitet

- Good quality code *avoids duplication*, displays *high cohesion*, *low coupling*.
- Coding style (*commenting*, *naming*, *layout*, etc.) is also very important.
- There is a big difference in the amount of work required to *maintain* poorly structured and well structured code.
- In fact, unless it is well structured, *the code is doomed* ... it will not be used for long.

Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling

16

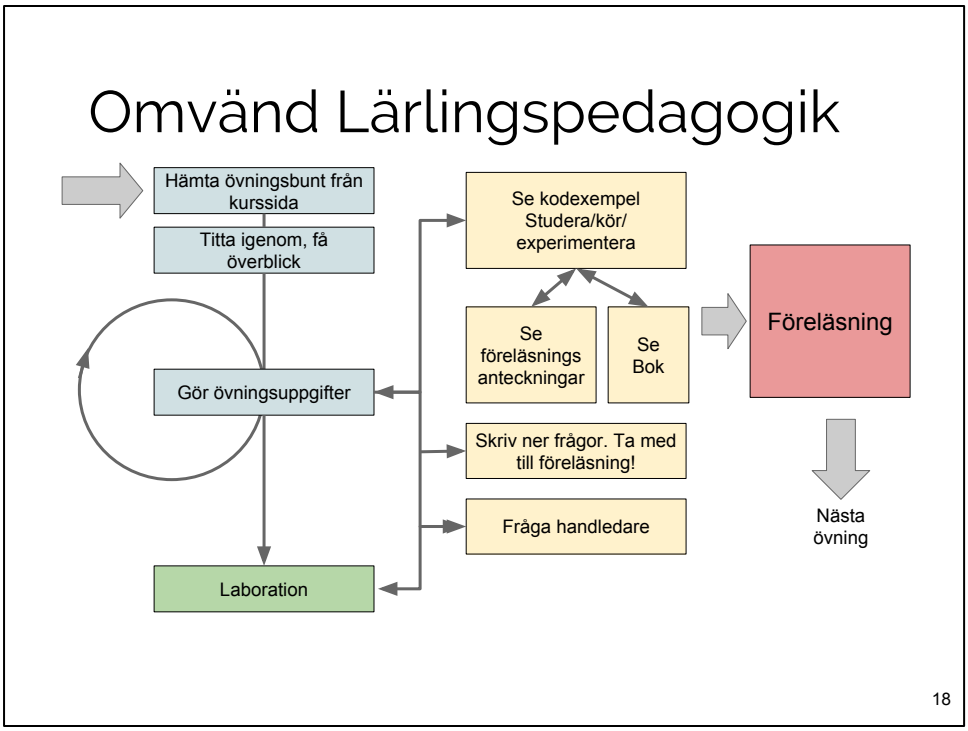
Förutom själva inlärningen av språket och kodandet sker det flera andra aktiviteter då man lär sig programmera

- Vi måste lösa problem som inte direkt har med kodandet att gör. Problemet "i sig" kan vara svårt.
- Vi måste arbeta på ett ordnat (ingenjörsmässigt) sätt
  - Om ej kör vi fast och kommer inte vidare!
- Våra program måste hålla av viss kvalitet, det räcker inte att programmet (kanske) fungerar.
- Vi bygger upp ett fackspråk

Att lära sig programmera 5

Du behöver papper och penna, … alltid!

- Visualisera, kommunicera!



Kursen använder "omvänd lärlingspedagogik"

- Omvänd = Kursen vänder på övningar och föreläsningar
  - D.v.s. ni börjar med övningarna utan att innehållet gåtts igenom på en föreläsning!
- Lärlings = Ni arbetar som lärlingar under övningarna och laborationerna , d.v.s. studerar (härmar) exempel och får handledning av en "mästare".

Varför?

- Genom att göra övningar först har ni "förarbetat" (tänkt till) inför föreläsningen. Lättare att hänga med och ställa frågor.
- Genom att ta med frågor från övningar/labbar har ni en chans att driva föreläsningar i den riktning ni vill.
  - Olika personer lär sig på olika sätt, ta chans att påverka!

Hur går övningen till?

- I början på övningen får ni en kort introduktion.
- Därefter gör man så många övningar man hinner (fortsätt gärna hemma)
- Hjälp får man genom att titta på exempel, se föreläsningsanteckningar och/eller bok, fråga handledare samt skriva ner frågor till föreläsningen.

- På slutet sammanfattar vi kort.

Efter övningen kommer en föreläsning i helklass
- Ta fram dina frågor, var aktiv!

Föreläsningsanteckningar

Föreläsningsanteckningar (bildserier) finns på kurssida

- Ordnade efter område
- Innebär att föreläsningar "hoppar runt" i olika serier.
  - Se vad som gäller för varje vecka på kurssidan
- Ev lite rörig i början men det blir enklare senare eftersom allt om ett visst område finns i en bildserie.
- Vissa bilder finns i flera serier

Inledningsvis dyker väldigt mycket upp

- Ni hinner inte smälta allt, härma så länge
- Vi repeterar och fördjupar efter hand (många gånger det som är svårt)

19

Jag kommer troligen inte att hinna gå igenom varenda bild

- Viktigt att gå igenom allt själv!

Tyvärr en bug i Google Docs: Sidnummer stämmer inte. Alla anteckningar börjar på sidan 1.

En formel ...

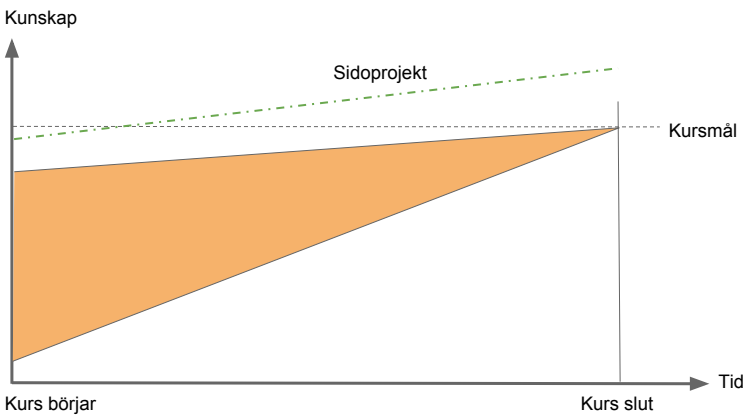
$$T_v = \pi T_b$$

20

Programmering är tidskrävande!

- En vanlig uppskattning: Verklig tid (Tv) är lika med beräknad tid (Tb) gånger pi.

Sidoprojekt



21

I inledande programmeringskurser är spridningen bland studenter mycket stor.

- För er som kan en del kan det bli lite för långsamt
  - Därför …. om intresse finns kan vi köra ett sidoprojekt parallellt.

Frågor

?