

Välkomna till TDA548

Grundläggande programvaruutveckling

```
private List<String> isValidPostfix(String[] tokens) {  
    List<String> output = new ArrayList<>();  
  
    Stack opStack = new Stack<>();  
    for (String token : tokens) {  
        if (Operators.isOperator(token)) {  
            opStack.push(token);  
        } else if (token.equals("(")) {  
            opStack.push(token);  
        } else if (token.equals(")")) {  
            output.add(opStack.pop());  
        } else {  
            output.add(token);  
        }  
    }  
  
    while (!opStack.isEmpty()) {  
        output.add(opStack.pop());  
    }  
    return output;  
}  
  
private List<String> checkOpStack(String op, Stack opStack, List<String> output) {  
    while (!opStack.isEmpty()) {  
        if (Operators.isOperator(opStack.peek())) {  
            break;  
        }  
    }  
    if (Operators.getPrecedence(op) < Operators.getPrecedence(opStack.peek())) {  
        output.add(opStack.pop());  
    } else if (Operators.getPrecedence(op) == Operators.getPrecedence(opStack.peek()) &&  
        Operators.getAssociativity(op) == Operators.Assoc.LEFT) {  
        output.add(opStack.pop());  
    } else {  
        break;  
    }  
}  
  
opStack.push(op);  
return output;  
}
```

Lärare och Handledare

Kursansvariga, examinatorer, föreläsare och handledare

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

Övriga Handledare (IT studenter)

- Robin Lilius-Lundmark
- Magnus Linderholm Wamby
- Edvin Meijer
- Gustav Lahti
- Abdullatif Alshriaf

Kursrepresentanter

Ni kan alltid tala med mig (JvH), vill ni inte det kan ni tala med representanterna (finns på kurssida)

- Robin Hallabro
- Ludwig Karlsson
- Alexander Larnemo Ask
- Otto Lundin
- Elin Nilsson

(Finns en kvalitetsprocess för kurser på Chalmers)

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
- Fel och oklarheter kan dyka upp, meddelas på kurssida

Innehåll

Vecka 1: Grunderna; variabler, tilldelning, selektioner, iterationer, metoder, inledande om array och klass m.m.

Vecka 2: Referenser m.m

Vecka 3: Typer m.m.

Vecka 4: Texter och samlingar (vad finns färdigt i Java)

vecka 5: Mer om klasser

Vecka 6: Objektorienterade tekniker. Mer om typer

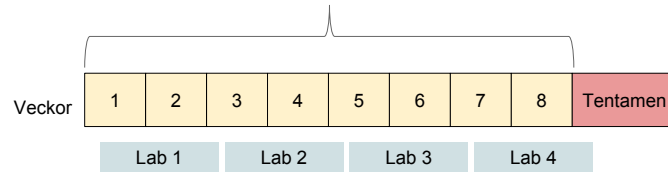
Vecka 7: Intensivträning inför tenta

Vecka 8: Tenta rep.

Se även kursplan, länk från kurssida.

Planering

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



Övningar och laborationer i datorsalar.

Grupper: A, B, C, D, se schema (anges ingen grupp får alla komma)

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
- **Interactive** (på schemat). Vi kodar övningar tillsammans ...

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.

7

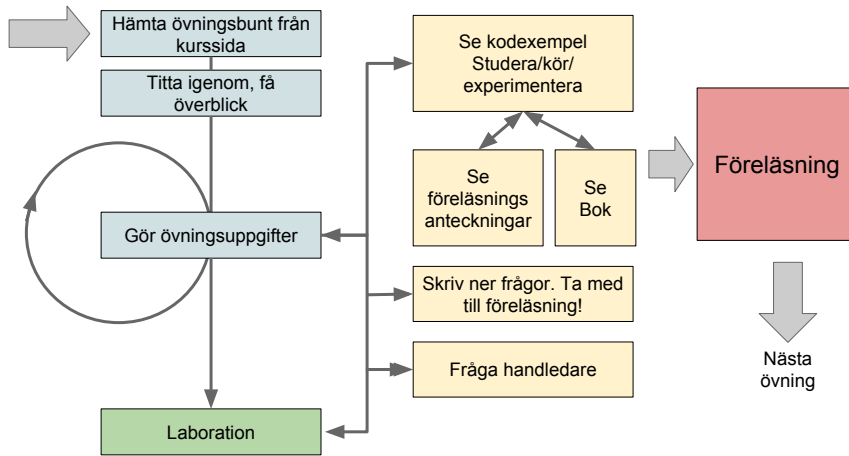
Lägg inte ut lösningar till övningar på nätet!

- Ni förstör för era kamrater!

Lägg inte ut lösningar till laborationer på nätet!

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

Omvänd Läringspedagogik



8

Kursen använder "omvänd läringspedagogik"

- Omvänd = Kursen vänder på övningar och föreläsningar
 - D.v.s. ni börjar med övningarna utan att innehållet gåtts igenom på en föreläsning!
- Lärings = 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!

Tentamen

Skriftlig tentamen (se exempel på kurssida >
Resurser)

- 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 ..
- .. och läs bildserier (slides)

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

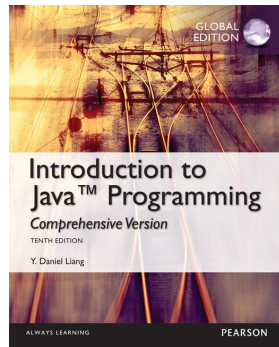
Språk

Ord/uttryck	Typ av ord	Engelskt ord som åsyftas	Förslag på översättning
A			
attach	v.	attach	bifoga
attachment	s.	attachment	bilaga
B			
backslash	s.	direkt inlån	bakåtvänt snedstreck, bakåtsreck, bakstreck ²
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 ²
cancel/cancel	v.	cancel	avbryta
cc:a	v.	direkt inlån	skicka kopia (till)
chatta	v.	to chat	prata, tala, småprata, snacka
Connect	v.	Connect/Connect to	Koppla upp, ansluta
D			
debugga	v.	to debug	avläsa, 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.

Bok



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 föreläsningssanteckningar.

Föreläsningsanteckningar

Föreläsningsanteckningar (bildserier) finns på kurssida

- Ordnade efter område (masterbilder)
- Veckobilerna är sammanställda utifrån områdebilderna
- Vissa bilder finns i flera serier

Inledningsvis dyker väldigt mycket upp

- Ni hinner inte smälta allt, härma så länge
- Se även minneskort på kurssidan > Föreläsningar
- Vi repeterar och fördjupar efter hand (många gånger det som är svårt)

13

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.

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 (JDK inte JRE)
- Därefter IntelliJ

Java 8 = Java 1.8

Att lära sig programmera 1



15

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

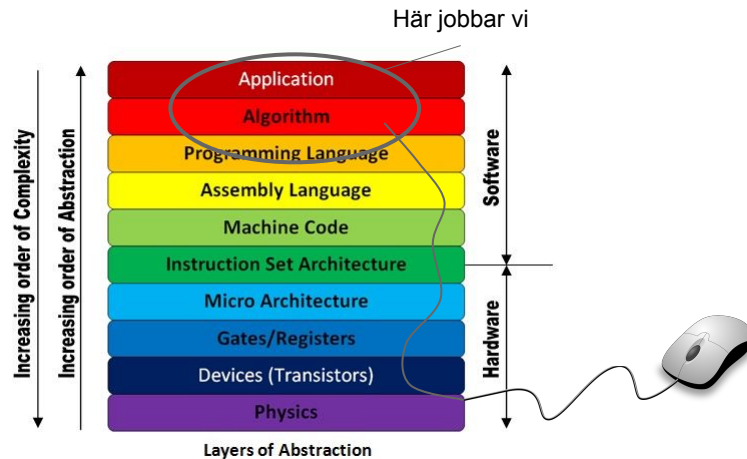


16

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 uppenbart.

Att lära sig programmera 3

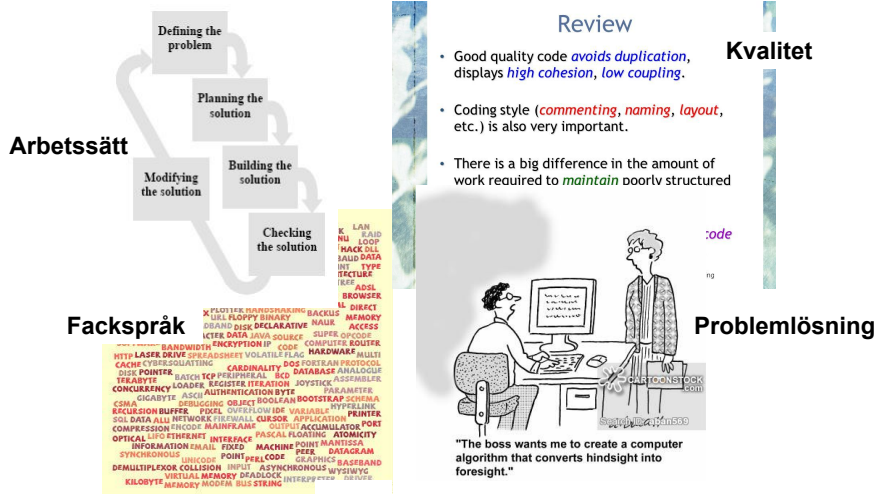


17

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

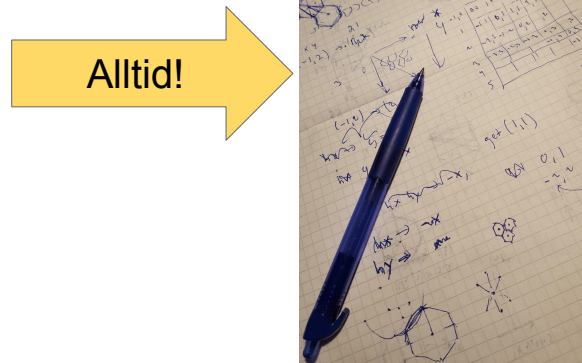


18

Förutom själva inläringen 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 göra. 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



19

Du behöver papper och penna, ... alltid!

- Visualisera, kommunicera!

En formel ...

$$T_v = \pi T_b$$

20

Programmering är tidskrävande!

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

Frågor

?