

Programmering

Seminarier i datavetenskap, datorteknik och informationsteknik

Niklas Broberg

niklas.broberg@chalmers.se

2015-09-24

Hur många från...

- Datavetenskap?
- Datateknik?
- Informationsteknik?

Översikt

- Kort feedback från introprogrammeringen
 - Fyll i enkäten!!
- Programmering som koncept
 - Vad är det? Vad omfattas rent generellt? Hur förhåller sig programmeringen till andra områden inom D&IT?
- Programmering i er utbildning
 - Vad kommer ni möta? Vad skiljer de tre studieprogrammen åt? Vilka möjligheter finns till fördjupning?

Quiz: Vanligaste svaret på frågan

”Vad har kursen lärt mig om mitt eget lärande?”

**Reflektera
(41)**

**Diskussion
är bra!
(19)**

**Jag kan mer
än jag tror
(10)**

**Våga fråga
(5)**

**Jobba aktivt med
materialet (17)**

**Studieteknik
(31)**

Grupparbete (11)

**Inget, jag
kunde allt (3)**

Noterbart

- "Att det är upp till mig själv om jag vill lära mig något"
- "Att hobbyprojekt är viktiga"
- "Att jag inte borde jämföra mig så mycket med andra"

Quiz: Vanligaste svaret på frågan

”Vad kommer jag minnas om fem år?”

**Python
(39)**

**Pyramiden
(24)**

Allt! (4)

**Tänka
logiskt (3)**

Reflektera (21)

**Studieteknik
(25)**

Uni /= gymnasiet (4)

Inget (1)

Noterbart

- ”De här lapparna” (3)
- ”Kunskapen om lärande är applicerbar bortom programmeringen”
- ”Jag kan lära mig vad som helst om jag bara tror på mig själv och försöker”

Quiz: Vanligaste svaret på frågan

”Vilka råd vill jag ge till nästa års studenter?”

Utmana er själva! (28)

Ta föreläsningarna på allvar (21)

Diskutera (16)

Försök själv hemma (11)

Hitta ”rätt” grupp (20)

Var aktiv (20)

Reflektera (8)

Våga fråga! (25)

Noterbart

- ”Lyssna – det är inte samma skit som tidigare”
- ”Man lär sig så mycket som man vill lära sig”
- ”Våga fråga – det är nu folk har som mest förståelse för din okunskap”

Programmering – vad är det?

- Kommunikation mellan programmerare och dator
- Konkret användning av ett programspråk, bibliotek, ramverk, ...
- Modeller, paradigmer, mönster, algoritmer, ...
- (Möjligheter och begränsningar, beräkningsbarhet, ...)



Abstraktion vs precision

- ”Hög-nivå vs låg-nivå”
 - Hur utvecklar vi program så effektivt som möjligt?
 - Hur utvecklar vi så effektiva program som möjligt?
- Abstraktion (”hög-nivå”) *kan* hjälpa till med båda
 - ... men ibland krävs precision för det senare (ex Datorgrafik)

Paradigmer

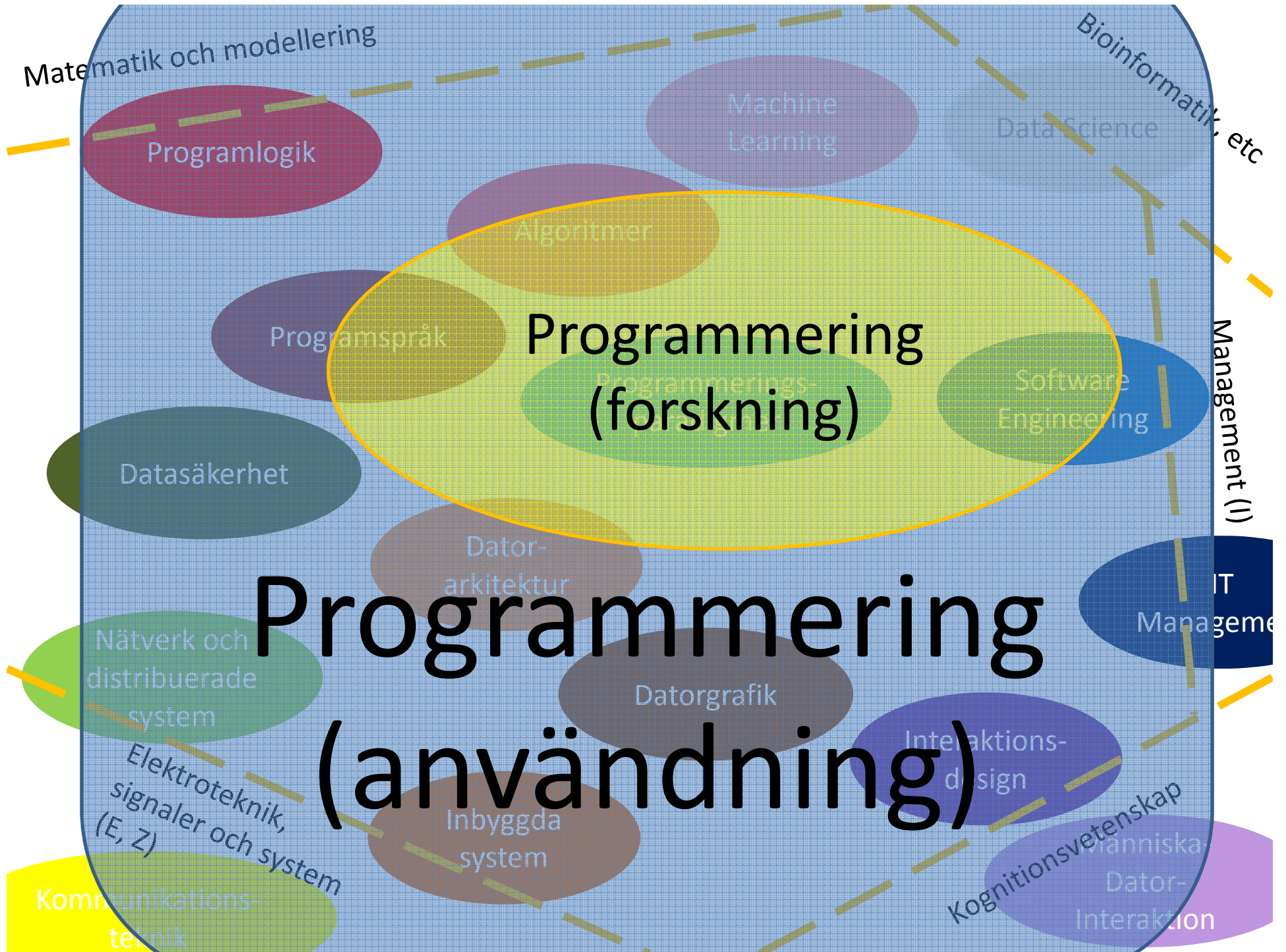
- Sätt att tänka kring programstruktur
- Objekt-orienterad, imperativ, funktionell, reaktiv, parallell, ...
 - Inga tydliga gränser
- Olika programspråk har inbyggt stöd för vissa paradigmer
 - ... men de flesta paradigmer går att applicera i de flesta språk – bara med olika mycket stöd

Mönster

- Givet en paradigm, hur strukturerar jag mitt program?
 - Inbyggda verktyg ("vad"): Funktioner, metoder, objekt, algebraiska datatyper, typklasser, generiska metoder, ...
 - Användningsmönster ("hur"): Events, observers, stream transformers, semaphores, factories, facades, ...
- Ett mönster går oftast att uttrycka i vilket språk som helst – det är bara mer eller mindre komplicerat
 - Olika språk kan ha inbyggt stöd för olika mönster

Forskning om programmering?

- Hur kan vi skriva mer effektiva program?
 - Algoritmer, programspråk, datorarkitektur
- Hur kan vi utveckla program mer effektivt?
 - Software engineering, programspråk
- Hur kan vi göra programmering lättare?
 - Programspråk, paradigmer



Vi utbildar inte programmerare!

- Vi utbildar datavetare och ingenjörer!
- Mycket bredare roller än ”programmerare”

Plats för alla!

- Alla behöver inte (bör inte!) bli programmerare!
- ... men alla behöver förstå programmering.

Tre studieprogram

- **Datavetenskap**
 - Väldigt brett, (extra) flexibelt program
 - Teoretiskt fokus
- **Datateknik**
 - DATAvetenskap + datorTEKNIK
 - Flexibelt – men spetsfokus mot hårdvara
- **Informationsteknik**
 - Flexibelt – men spetsfokus mot mjukvara

Programmering i utbildningen

- Vi tillhandahåller tre tydliga spår:
 - Objekt-orienterad programmering
 - Maskinnära programmering
 - Funktionell programmering

Objektorientering (OO)

- OO programmering (introduktionskurs)
 - Grundläggande imperativ programmering, introduktion till objekt och OO-tänkande (Java)
- OO programmering, fortsättningskurs
 - Stort fokus på OO-tänkande
- OO programmeringsprojekt
 - Applicera i praktiken
- (Modellering, processer, Software Engineering (Master))

Maskinnära programmering

- Grundläggande Datorteknik
 - Introduktion till lågnivå-programmering, assembler
- Programmering av Inbyggda System
 - Fokus på maskinnära programmering (C)
- Datatekniskt projekt
 - Applicera i praktiken
- (Inbyggda system (Master), datorsystem, hårdvarukonstruktion ...)
- Mer nästa seminarium: Digital konstruktion och inbyggda system, 8/10 (Lp1 w6)

Funktionell programmering (FP)

- (Introduktion till) FP
 - Grundläggande funktionell programmering, introduktion till FP-tänkande
- Avancerad Funktionell Programmering
 - Stort fokus på FP-tänkande
- (Parallell FP)
- Mer på seminariet om Funktionell programmering, 22/10 (Lp1 w8)

Breda koncept

- Datastrukturer, Algoritmer
 - Applicerbart inom alla paradigmer (om än med viss specialisering)
 - Läses av alla programmen
 - Mer om detta på seminariet om Algoritmer (Lp2, w1)
- Parallell programmering
 - Arbeta med trådar, processer och parallellism

Bredd vs spets

- Bredd: Läs alla introduktionskurser!
 - Ger en större förståelse än summan av komponenterna
- Spets: Fokusera på ett område för framtida specialisering
 - Största skillnaden mellan programmen

Datavetenskap

- LP1: Introduktion till Funktionell Programmering (DIT440)
- LP2: Imperativ programmering med grundläggande objektorientering (DIT012)
 - Första OO-kurs
- LP3: Objektorienterad programmering och design (DIT952)
 - Andra OO-kurs
- År 2/3: Parallell programmering (DIT390); massor av valbart!
 - Ex: Grundläggande datorteknik (DIT791); Programmering av inbyggda system (DIT152); Advanced FP (DIT260);

Datateknik

- LP1: Introduktion till Funktionell Programmering (TDA555)
- LP2: Grundläggande Datorteknik (EDA452)
 - Första maskinnära kurs (assembler)
- LP3: Objektorienterad programmering (DAT043)
 - Första OO-kurs
- LP4: Programmering av inbyggda system (EDA481)
 - Andra maskinnära kurs (C)
- År 2/3: Datatekniskt projekt (DAT290); valbart.
 - Ex: Parallell Programmering (TDA383); Advanced FP (TDA342)

Informationsteknik

- LP1: Objektorienterad programvaruutveckling (TDA545)
 - Första OO-kurs
- LP1: Grundläggande datorteknik (EDA433)
 - Första maskinnära kurs (assembler)
- LP2: Objektorienterad programvaruutveckling, fortsättningskurs (TDA550)
 - Andra OO-kurs
- LP4: Objektorienterat programmeringsprojekt
 - Tredje OO-kurs
- År 2/3: Programmering av inbyggda system (DAT016); valbart.
 - Ex: Parallell Programmering (TDA383); Funktionell Programmering (TDA452); Advanced FP (TDA260);

Vidare möjligheter

- Mer i kommande seminarier om:
 - Inbyggda system (nästa seminarium)
 - Fokus på maskinnära program
 - Precision, hårdvaruintegrering
 - Programspråksteknologi (lp3)
 - Teorier och mekanismer bakom programspråk
 - Software Engineering (lp4)
 - Fokus på mjukvaruutveckling
 - Modeller, högre-nivå-abstraktion
 - Processer för utveckling

Frågor?

Digital konstruktion och inbyggda system

- Nästa seminarium:
 - Vem: Lars Svensson
 - När: 2015-10-08
 - Var: HB4

Vart vi än vänder oss omges vi av datoriserade manicker: på skrivbordet, på soffbordet, på väggen i vardagsrummet, i fickan, i bilen... Och detta är bara de som vi ser på nära håll: varken telenätet, elnätet, vattenförsörjningen, transportsystemen eller det mesta av industriproduktionen skulle fungera utan inbyggda elektronik- och datorsystem, i alla fall inte effektivt.

Området "Digital konstruktion och inbyggda system" handlar om alla dessa manicker och om hur man konstruerar dem så att de erbjuder nödvändig flexibilitet, prestanda och batteritid. Vi rör oss från enstaka transistorer och grindar upp till enklare processorkärnor, där området "Datorarkitektur" tar vid; men även mot områden där prestanda- och effektkrav är alltför tuffa för processorlösningar.