

Högskoleverkets kvalitetsutvärderingar 2011 – 2014

Självvärdering

Chalmers tekniska högskola

Område för yrkesexamen: Datateknik

Examen: Civilingenjör

Utvärderingsärende reg.nr: 643-01844-12



Innehåll

| | |
|---|----|
| Innehåll | 2 |
| Introduktion..... | 3 |
| Examensmål 0..... | 3 |
| Del 1 | 8 |
| Examensmål 1 | 8 |
| Delmål 1a: kunskap om det valda teknikområdets vetenskapliga grund och beprövade erfarenhet | 8 |
| Delmål 1b: insikt i aktuellt forsknings- och utvecklingsarbete | 9 |
| Examensmål 2..... | 11 |
| Delmål 2a: Brett kunnande inom det valda teknikområdet, inbegripet kunskaper i matematik och naturvetenskap | 11 |
| Delmål 2b: väsentligt fördjupade kunskaper inom vissa delar av det valda teknikområdet..... | 12 |
| Examensmål 3..... | 16 |
| Delmål 3a: Förmåga att med helhetssyn kritiskt, självständigt och kreativt identifiera, formulera och hantera komplexa frågeställningar och därigenom bidra till kunskapsutvecklingen | 17 |
| Delmål 3b: Förmåga att delta i forsknings- och utvecklingsarbete och därigenom bidra till kunskapsutvecklingen..... | 18 |
| Examensmål 4..... | 21 |
| Delmål 4a: Förmåga att utveckla och utforma produkter, processer och system | 21 |
| Delmål 4b: Förmåga att ta hänsyn till människors förutsättningar och behov och samhällets mål för ekonomiskt, socialt och ekologiskt hållbar utveckling | 21 |
| Examensmål 5..... | 23 |
| Delmål 5a: Förmåga att muntligt klart redogöra för och diskutera sina slutsatser och den kunskap och de argument som ligger till grund för dessa..... | 23 |
| Delmål 5b: Förmåga att skriftligt klart redogöra för och diskutera sina slutsatser och den kunskap och de argument som ligger till grund för dessa..... | 24 |
| Delmål 5c: Förmåga till dialog med olika grupper..... | 25 |
| Examensmål 6..... | 26 |
| Software Engineering..... | 26 |
| Avslutande slutsatser om måluppfyllelse – del 1 | 28 |
| Del 2 | 29 |
| Lärarkompetens och lärarkapacitet | 29 |
| Antal helårsstudenter | 29 |
| Analys och slutsats | 29 |
| Del 3 | 30 |
| Andra förhållanden | 30 |
| Om examensarbetet | 30 |
| Bilaga: tabell – Lärarkompetens och lärarkapacitet | 34 |

Introduktion

Vi lever i en oerhört spännande tid när datatekniken infiltrerar snart sagt varje del av samhället. Vi är alla användare av datatekniska produkter och tjänster och det är svårt att tänka sig vårt moderna samhälle utan mobiltelefoner, Internet och ABS-bromsar. För användare räcker det att tekniken fungerar (och ser bra ut) men i bakgrunden måste någon planera, utveckla, implementera och underhålla de ofta programvarubaserade systemen. Det är där som huvudfokus är för Datateknikutbildningen nu och i framtiden. Samhället (inklusive industrin) behöver dataingenjörer som har koll på god kvalitet, hög produktivitet och låga kostnader.

Men en högre utbildning är så mycket mer än så - personlig utveckling, nyfikenhet och nya idéer är också viktiga delar. Många av våra alumni jobbar inte bara med det de lärt sig på Chalmers men en sund bakgrund i teori och problemlösning räcker långt inom nästan alla områden¹. Många går vidare till forskning och utveckling och de nämner ofta inspiration från kurser och utmaningar redan tidigt i utbildningen.

Slutligen finns också övergripande utmaningar för mänskligheten i stort som delproblem inom den globala omställningen till en hållbar utveckling. Tsunamivarningar, klimatsimuleringar och smart energiförsörjning är bara några exempel. Här finns det mycket att göra med datateknik som verktyg och därmed passar utbildningen i datateknik väl in i visionen "Chalmers - för en hållbar framtid".

De första åren ger studenterna flera nya språk för att formulera sina tankar och strukturera olika lösningar. Det matematiska språket, flera olika språk för algoritm- och systemdesign samt en förståelse för fysikens och datateknikens begränsningar och möjligheter. Redan tredje året ger stor möjlighet att välja kurser. De två sista åren genomförs i form av masterprogram där studenterna specialiserar sig tillsammans med andra svenska och utländska studenter som delar deras intressen. Varje masterprogram har stor valfrihet, vilket förstärker möjligheten att individualisera sin utbildning.

Examensmål 0

Det övergripande målet för civilingenjörsexamen är att studenten ska "*visa sådan kunskap och förmåga som krävs för att självständigt arbeta som civilingenjör*". För Chalmers i allmänhet och för Datateknikutbildningen (D) i synnerhet nås detta mål mycket väl. Av de med civilingenjörsexamen från D 2009 som svarat på Chalmers alumnenkät 2012 hade hela 75 % fått sitt första arbete som var relevant med avseende på sin utbildning redan *före* examen. Ytterligare 17 % hade fått relevant jobb inom ett halvår efter examen. Drygt 10 % av studenterna går vidare till doktorandstudier lokalt, nationellt eller internationellt. D-alumni har efter tre år lön mellan 30 och 40 kkr och de är genomgående nöjda med sin utbildning (alla som svarat ger betyg 7–10 av 10 möjliga och medianen är 8).

Termen Datateknik (D) används i detta dokument som motsvarande det breda begreppet Computer Science and Engineering (CSE). Huvudområdet Datateknik på Chalmers innefattar civilingenjörsexamen, masterexamen, kandidatexamen och högskoleingenjörsexamen. De tre första (alla utom högskoleingenjörsexamen) nås huvudsakligen genom olika delar av samma utbildningsprogram: Datateknik, 300hp. Civilingenjörsutbildningen är femårig, men den är sedan 2005 organiserad i en treårig kandidatdel och en tvåårig masterdel. Av de som slutför D-utbildningen tar ca 10 % ut enbart en kandidatexamen, övriga tar ut en masterexamen och en civilingenjörsexamen. Civilingenjörerna från D på Chalmers fördelar sig på fem olika masterprogram med både nationell och internationell rekrytering. Dessa program finns inom tre olika närliggande (och delvis överlappande) huvudområden:

¹ Förutom kärnan inom IT-branschen (Ericsson, konsultföretag) har vi har alumni inom exempelvis fastighetsbolag, ICA, och Nordea.

elektroteknik (E), datateknik (D), samt informationsteknik (IT). Det senaste läsårets studentfördelning visas nedan där programmen listas i ämnesordning från ”hårdare” till ”mjukare”.

| Programnamn med länk till programbeskrivning (kortform) | Andel | Huvudområde |
|---|--------------|--------------------|
| Embedded Electronic System Design (EESD) | 12 % | E |
| Computer Systems and Networks (CSN) | 26 % | D |
| Computer Science—Algorithms, Languages and Logic (CS-ALL) | 37 % | D |
| Software Engineering (SE) | 12 % | IT |
| Interaction Design and Technologies (ID&T) | 12 % | IT |

D-området består (något förenklat) av tre delar: en gemensam bas av obligatoriska kurser under de första tre åren, en valbar del (fem valbara kurser) på kandidatnivå och ett val bland några ackrediterade masterprogram (som i sin tur har ett obligatorium och en valfri del). Valfriheten gör det möjligt för studenter inom D-området att sätta en personlig prägel på sin examen. Detta är bra för individens motivation och för att hålla examen aktuell men också för att uppfylla examensordningens tolfte mål: *“visa förmåga att identifiera sitt behov av ytterligare kunskap och att fortlopande utveckla sin kompetens”*. I alumnenkäten ger studenterna själva betyg 9 av 10 möjliga på just detta mål (högre än Chalmers i övrigt).

Datateknikområdets mål och innehåll. D-området (här representerat av civilingenjörsprogrammet D, 300hp) har 12 mål som är specialiseringar av examensordningens mål. Motsvarigheten till de sex mål som nu utvärderas finns (på engelska) i figur 1 och alla målen finns i Utbildningsplanen i Chalmers studentportal. Målen är framtagna på engelska eftersom många av studenterna (och vissa av lärarna) på masternivå inte kan svenska. Under 2011 hölls två lärarmöten och ett programrådsmöte just med syftet att ta fram dessa mål på programnivå, förankra dem i lärarkollegiet, studentgruppen och bland industrirepresentanterna samt koppla dem till individuella kursers lärandemål och examination.

Tabell 1: De 6 av D-områdets 12 mål som utvärderas (numrerade enligt HSV-utvärderingen).

D mål 1: Based on proven experience as well as insights of current research and development, the programme graduate should be able to demonstrate knowledge in:

Hardware / software interaction: explain, model and demonstrate how to control computer systems using existing and future computer hardware and other computational manners.

Computer programming and the software development processes.

Theory of computation and inter-computer interaction: algorithms, complexity, intractability, logic of a single and many computational units.

Hardware design and development process.

D mål 2: demonstrate broad knowledge in software and hardware, including mathematics and science areas connected to the CSE area, as well as significant depth of knowledge in certain parts of the area (the chosen specialisation / MSc programme).

D mål 3: demonstrate the ability to with a holistic view identify, formulate and handle complex issues in a critical, independent and creative manner, and be able to participate in research and development and thereby contribute to the development of the field. Holistic view here means: Both the hardware- and software-aspects of CSE as well as their economic, organisational and sustainable aspects. Strong abilities within engineering science, natural science and some abilities within social science.

D mål 4: Demonstrate engineering and professional abilities to model, develop and design computer-related products, software development processes and computer systems, which take into consideration human capacity, the society goals and sustainable developments. In particular, technical and professional skills related to mathematical, engineering, inter-personal communication, leadership and ethics.

D mål 5: Demonstrate the ability to discuss and give a clear account of his/her conclusions and the knowledge and arguments that support these both orally and in writing addressing different audiences in both national and international contexts.

Groups: Other engineers in the same subject, engineers in nearby subjects, to some degree stakeholders / customers.

International: International campus, international groups, papers for international contexts.

D mål 6: Demonstrate insights of the potential and limitations of information and communication technology, its role in society and the responsibility of people for how it is used, with regards to social, economical and environmental aspects including the working environment.

De tre åren som leder till kandidatexamen innehåller 17 * 7.5hp obligatoriska kurser, 5 * 7.5hp valfria kurser och ett avslutande kandidatarbete om 15hp (se översikten i figur 2). Inom det valfria blocket på kandidatnivå tog studenterna:

- [Parallell programmering](#) eller [Programspråk](#), och
- [Databaser](#) eller [Grundläggande software engineering](#), samt
- en masterkurs: [Algorithms](#), [Computer Graphics](#), [Cryptography](#) eller [Artificial Intelligence](#).

De sista två kurserna på kandidatnivå som studenterna tar är spridda över en större grupp kurser både inom D-området och inom Människa, Teknik, Samhälle (MTS).

På masternivån fortsätter två tredjedelar av studenterna till masterprogrammen CS-ALL eller CSN (som också fungerar som inriktningar på civilingenjörsprogrammet) där de blandas med lika många till med annan bakgrund men samma mål: fördjupade studier i Datateknik. Där ingår 3-4 obligatoriska kurser, 7-8 valfria kurser och ett examensarbete, ofta i nära samverkan med industrin. Studenterna erbjuds valfria kurser inom åtta spår: Algoritmer, Logik och verifiering, Programspråk, Datasäkerhet, Parallella och distribuerade system, Nätverk och kommunikation, Datorsystemteknik, samt Programmering av datorsystem.

En tredjedel av studenterna fortsätter till något av de närliggande masterprogrammen EESD, SE och ID&T för att fördjupa sig inom Inbyggda system, Software Engineering eller Interaktionsdesign. Dessa tre program är lite mer styrda: 45hp obligatorium, 45hp valfritt och 30hp examensarbete. Jämfört med huvudfåran för D-ingenjörerna erbjuds här en större bredd och djup inom hårdvara och datorarkitektur, programvaruutveckling och dess processer, eller människa-dator interaktion och spel.

Hela det breda utbudet av obligatoriska och valfria kurser inom dessa fem program är tillgängligt som valfria kurser för alla D-studenter som uppfyller förkunskapskraven. Detta innebär att studenterna kan skraddarsy sin egen kompetensprofil och på så sätt själva optimera sina möjligheter att efter examen självständigt arbeta som civilingenjör.

En genomgående princip för väldigt många av D-områdets kurser är en iterativ process av inlämning och återkoppling. Ofta är det i form av laborationer där studenterna löser en uppgift och lämnar in programkod i någon form, får återkoppling och sedan lämnar in förbättringar tills de blir godkända. Och även mellan inlämningarna finns det oftast en snabbare cykel av automatisk återkoppling från kompilatorer eller testramverk som också främjar lärandet. Eller som Piet Hein formulerar det:

THE ROAD TO WISDOM?

*Well, it's plain
and simple to express.*

*Err
and err
and err again,
but less
and less
and less.*

– Piet Hein, Grooks, MIT Press, 1967.

Tabell 2: Programöversikt D civilingenjör. Procenttalen visar andelen studenter med en civ.ing.-examen som har läst den kursen eller det masterprogrammet.

| Kursnamn | Andel | Kommentar |
|--|-------|-------------------|
| Introduktion till funktionell programmering | | År 1, obl. |
| Inledande diskret matematik | | År 1, obl. |
| Digital- och datorteknik | | År 1, obl. |
| Linjär algebra | | År 1, obl. |
| Maskinorienterad programmering | | År 1, obl. |
| Matematisk analys | | År 1, obl. |
| Bärkraftig resursanvändning | | År 1, obl. |
| Elektriska kretsar och fält | | År 1, obl. |
| Objektorienterad programmering | | År 2, obl. |
| Matematisk statistik och diskret matematik | | År 2, obl. |
| Datastrukturer | | År 2, obl. |
| Fysik för ingenjörer | | År 2, obl. |
| Digitalteknik-syntes | | År 2, obl. |
| Datorsystemteknik | | År 2, obl. |
| Datakommunikation | | År 2, obl. |
| Parallell programmering | 77 % | År 2-3, en av två |
| Programspråk | 68 % | År 2-3, en av två |
| Transformers, signaler och system | | År 3, obl. |
| Reglerteknik | | År 3, obl. |
| Kandidatarbete, 15hp | | År 3, obl. |
| Databaser | 78 % | År 3, valfri |
| Grundläggande software engineering | 52 % | År 3, valfri |
| Algorithms (MSc) | 66 % | År 3-5, valfri |
| Cryptography (MSc) | 45 % | År 3-5, valfri |
| Logic in Computer Science (MSc) | 44 % | År 3-5, valfri |
| Programming Paradigms (MSc) | 44 % | År 3-5, valfri |
| Artificial Intelligence (MSc) | 40 % | År 3-5, valfri |
| Computer Graphics (MSc) | 37 % | År 3-5, valfri |
| Embedded Electronic System Design | 12 % | Masterprog. 120hp |
| Computer Systems and Networks | 26 % | Masterprog. 120hp |
| Computer Science—Algorithms, Languages and Logic | 37 % | Masterprog. 120hp |
| Software Engineering | 12 % | Masterprog. 120hp |
| Interaction Design and Technologies | 12 % | Masterprog. 120hp |

Del 1

Examensmål 1

Kunskap och förståelse:

För civilingenjörsexamen skall studenten visa kunskap om det valda teknikområdets vetenskapliga grund och beprövade erfarenhet samt insikt i aktuellt forsknings- och utvecklingsarbete

Här liksom under flera senare mål delar vi upp argumentationen under delmål som matchar de kriterier som HVSs bedömargrupper enats om. Det är dock rätt ofta som samma moment i utbildningen passar in under flera rubriker (ibland under flera examensmål) så matchningen är inte perfekt. Vi hoppas att texten ändå blir läsbar för er som granskare.

Delmål 1a: kunskap om det valda teknikområdets vetenskapliga grund och beprövade erfarenhet

Det mest direkta måttet på måluppfyllelse är att D-studenterna är så eftertraktade på arbetsmarknaden – inom både industri och akademi. Det är kanske inte uppenbart att alla arbetsgivare efterfrågar just vetenskaplig grund och beprövad erfarenhet, men de som sitter i programrådet och övriga vi varit i kontakt med håller alla med om det. Under de obligatoriska kurserna i början av utbildningen skaffar sig studenterna en god överblick över D-området. Områdets fulla bredd beskrevs redan tidigare (sid 7) i form av de kurser och masterprogram som ingår, så vi nöjer oss här med två stickprov som beskriver lärssekvenser inom inbyggda system och inom programvaruteknik och software engineering.

Studenterna lär sig om inbyggda system (kopplingen mellan hårdvara och programvara) i en lärssekvens som bland annat kopplar samman kurserna i digital- och dator teknik, maskinnära programmering, datorsystemteknik, datakommunikation, reglerteknik och operativsystem med masterprogrammen CSN och EESD. Ett intressant exempel på ”beprövad erfarenhet” är att nästan alla studenterna i samband med kursen i Datakommunikation genomför en certifieringskurs: Cisco Certified Network Associate (CCNA) Exploration 4.0.

Studenterna lär sig programvaruteknik och software engineering genom en lärssekvens som bland annat kopplar samman kurserna i Funktionell programmering, Objektorienterad programmering, Datastrukturer, Grundläggande software engineering, Parallell programmering, Programspråk och Databaser med masterprogrammen CS-ALL och SE. En grundidé inom det här spåret är att studenterna skall bli ”allätare” när det gäller programspråk. Förändringstakten inom IT-världen är sådan att de antagligen kommer att behöva lära sig andra språk när de kommer ut och i större sammanhang behövs ändå flera samverkande språk. D-studenterna får en ordentlig dos av Haskell i År 1 Läspanod (Å1L1), C (Å1L3), Java (Å2L1), smakar på Matlab (Å1L2), SmallTalk (Å2L3), Erlang (Å3L1) och SQL (Å3L2). Programmeringsuppgifter med ökande svårighetsgrad bygger upp den språktekniska kompetensen och utvecklingsverktyg, utvecklingsprocesser och projektarbeten gör studenterna till mycket kvalificerade ”software engineers” när de tar examen.

När det gäller beprövad erfarenhet erbjuder vi sedan flera år tillbaka ett (frivilligt) mentorprogram: MEDIT = mentorprogrammet på Elektro, Data och Informationsteknik. Under studenternas sista år på programmet har de möjlighet att få en mentor från näringslivet. Mentorn bidrar med en erfarenhets- och kunskapsöverföring som är ovärderlig för studenten. Samtidigt utvecklas både student och mentor personligen och utökar sitt nätverk. Programmet är mycket uppskattat av dem som varit med.

Kommentarer från studenter: "Det som jag framför allt tyckte var bra med att ha en mentor var att man fick se sig själv med andras ögon", "Det är så givande!", "Bra initiativ!"

Kommentar från mentorer: "Gör det! Det ger massor!", "Tack för en rolig och lärorik resa!", "Det var mycket givande för mig personligen".

Hemsida: <http://www.mentoredit.se>

Delmål 1b: insikt i aktuellt forsknings- och utvecklingsarbete

Redan från start får Chalmers D-studenter kontakt med lärare som inte bara är pedagogiskt skickliga utan också är världsledande forskare. Ett exempel är första kursen i programmering där Koen Claessen introducerar funktionell programmering i Haskell och automatisk testning med QuickCheck. Koen är mottagare av både [Chalmers pedagogiska pris 2012](#) för den kursen och av "[Most Influential ICFP Paper Award 2010](#)" för artikeln som introducerade testningsverktyget QuickCheck för över tio år sedan. (ICFP är den mest kända internationell konferensen inom funktionell programmering.) Enligt juryn får Koen det pedagogiska priset

"... för sitt arbete med att reformera en stor inledande kurs med kraftig variation i förkunskaper och föreställningar om relevans bland studenterna. Han stimulerar till aktiva studier tidigt i kursen bland annat genom att använda kontinuerlig examination med snabb återkoppling. Genom att knyta an till såväl studenternas vardagsföreställningar som till kommande kurser och industrins efterfrågan upplever studenterna att den reformerade kursen har en hög relevans."

Från [kursens lärandemål](#) kan vi se att studenterna både lär sig om D-områdets vetenskapliga grund (modellering med datatyper, rekursion) och om tillämpliga metoder (användning av högre ordningens funktioner, testning med hjälp av lämpliga verktyg). Studenterna examineras dels kontinuerligt i form av laborationsuppgifter och dels en tentamen där för godkänt krävs 4 av 5 tillräckligt väl behandlade problem. Labuppgifterna innehåller effektiv beräkning av x^n , en förenklad version av spelet Black Jack, en Sudoku-lösare och en grafisk miniräknare.

Ett exempel i andra året är kursen i Parallell programmering där studenterna förutom ämnet i sig får lära sig om flera vinnare av Turing-priset och se exempel på forskningsartiklar vars resultat var banbrytande på sin tid men som nu är en normal del av kurslitteratur och övningar. Med detta som bakgrund och motivation presenteras också flera resultat av aktuell forskning i form av gästföreläsningar (Erlang, Software Transactional Memory).

Som exempel från tredje året passar kursen i Programmeringsparadigmer som är nyutvecklad och som baseras på resultat från Chalmers och GU:s långa historia av framgångsrik forskning om programspråk och språkteknologi. Idén för kursen är översättning mellan olika paradigmer: hur skriver man parallella program i ett imperativt programspråk? Eller hur simulerar man objekt i ett funktionellt språk? Tanken är att man skall lära sig de olika paradigmen bättre genom att jämföra, kontrastera, simulera och analysera. Studenterna får stifta bekantskap med Haskell, Erlang, Prolog och SmallTalk under övningar och på skriftlig tentamen.

Under de avslutande två årens masterprogram blir kopplingen till forskning och utveckling tätare. Här är nästan alla lärare aktiva forskare och flera är ledande experter i sina områden. D&IT-institutionen är välkänd för sin forskning inom funktionell programmering, testning, kombinatorisk, språkbaserad säkerhet, typteori, datorstödd teorembevisning, språkteknologi, algoritmer för nätverksanalys, flerkärniga datorsystem mm. Några exempel på stora anslag som nyligen har beviljats: John Hughes inom funktionell programmering och testning, Per Stenström inom datorarkitektur och parallellisering, Thierry Coquand inom typteori och David Sands inom datasäkerhet och programspråk. Dessa personer medverkar aktivt i programmets kurser på såväl grund som avancerad nivå som examinatorer, kurs-

ansvariga och gästföreläsare: John inom Parallell funktionell programmering och Programmeringsparadigm, Per inom Datorarkitektur, Thierry inom Logic in Computer Science och Types for Programs and Proofs, samt Dave inom Programspråk, Funktionell programmering och Datorsäkerhet. Forskningsanknytningen håller utbildningen i synk med den pågående utvecklingen av datateknikämnet och är en bas för ständig förnyelse. Se även under Delmål 3b (sida 18) för fler exempel på forskningsanknytning.

Sammanfattningsvis kan vi utan tvekan hävda att en civilingenjör från D på Chalmers står på en stadig vetenskaplig grund, har tagit till sig väl valda delar beprövad erfarenhet samt fått mycket god insikt i aktuellt forsknings- och utvecklingsarbete.

Examensmål 2

Kunskap och förståelse:

För civilingenjörsexamen skall studenten visa såväl brett kunnande inom det valda teknikområdet, inbegripet kunskaper i matematik och naturvetenskap, som väsentligt fördjupade kunskaper inom vissa delar av området

För D är ”det valda teknikområdet” datateknik och den ämnesmässiga bredden får studenterna huvudsakligen genom att programmets obligatoriska kurser spänner upp D-området från matematik och fysik, via elektriska kretsar, reglerteknik, digitalteknik, datorsystemteknik, programmeringsteknik, datakommunikation till bärkraftig resursanvändning. Väsentlig fördjupning uppnås till viss del genom de valbara kurserna och kandidatarbetet under tredje året men framför allt genom de två årens sista studier på masternivå.

Nedan beskriver vi hur Datateknik-studenter på Chalmers uppnår den del av målen som handlar om matematik. Det innebär också att avsnittet till viss del försöker definiera hur vi tolkar vilken matematik som är relevant och väsentlig för en blivande dataingenjör. (Detta eftersom det skiljer sig en del från en klassisk ingenjörstolkning.)

Delmål 2a: Brett kunnande inom det valda teknikområdet, inbegripet kunskaper i matematik och naturvetenskap

Det naturvetenskapliga innehållet i D-utbildningen beror på definitionen av termen ”naturvetenskap”. Med en tillräckligt smal tolkning är det bara kursen Fysik för ingenjörer som ingår och med en vidare tolkning ingår även Elektriska kretsar och fält, Reglerteknik och Bärkraftig resursanvändning. D-studenternas kunskaper inom dessa ämnen, samt inom matematiken, fungerar som en brygga till andra discipliner och ger en viss minsta gemensamma nämnare mellan olika ingenjörsutbildningar. Det bör dock nämnas att situationen är på väg att förändras – på samma sätt som matematiken alltid varit en grundbult i all ingenjörsutbildning börjar datateknik och beräkningsteknik blir mer och mer det ”lingua franca” som förenar ingenjörer från alla ämnen.

Matematiska metoder är en viktig del av D-utbildningen och ingår i nästan alla kurser. Den gemensamma basen innehåller fyra ”rena matematikkurser” á 7,5 hp. Det D-specifika är tonvikten på diskret matematik redan från början:

- Inledande diskret matematik,
- Linjär algebra,
- Matematisk analys
- Matematisk statistik och diskret matematik.

Dessutom finns inslag av matematik och naturvetenskap i flera av de övriga kurserna inom den gemensamma basen; i samråd med Samuel Bengmark (Matematiska vetenskaper, Chalmers) har vi identifierat ytterligare 30 hp ”matematiska ämnen” (av varierande storlek från 1,5 hp till 6 hp per kurs) inom följande obligatoriska kurser:

- Introduktion till matematik,
- Introduktion till funktionell programmering,
- Elektriska kretsar och fält,
- Objektorienterad programmering,

- Datastrukturer,
- Fysik för ingenjörer,
- Transformer, signaler och system,
- Kandidatarbete, 15hp.

I den valfria delen av utbildningen kan D-studenterna välja mer matematik och de mest populära på kandidatnivå är Matematisk modellering samt Flervariabelanalys. På masternivån finns det många möjligheter att fördjupa sig matematiskt inom beräkningsbarhet, algoritmer, komplexitet, optimering, logik, bevis, formella språk, funktionell programmering, mm. Man kan bli civilingenjör från D även med masterprogrammet ”Engineering Mathematics and Computational Science” som avslutning inom huvudområdet Matematik, men inom just det utvärderade läsåret tog ingen student den vägen.

Vi fokuserar på noggrann modellering, analys och problemlösning: exempelvis är uppgifter och tentamensfrågor i de algoritmiska kurserna huvudsakligen baserade på matematisk problemlösning. Andra exempel:

- Kursen ”Cryptography” illustrerar att talteori (som bara varit en intellektuell lek under århundraden) har ett mycket praktiskt värde som teoribas för autentisering och krypteringsmetoder. Studenterna förväntas beskriva och analysera dessa talteoretiskt grundade metoder.
- Kursplanen för ”Types for Programs and Proofs” säger bland annat “Examples will be given which illustrate the unity of mathematics and programming in this theory”. Studenternas inlämningsuppgifter och presentationer visar att de lär sig både hur man gör funktionell programmering i ett språk med beroende typer och hur man matematiskt bevisar egenskaper hos språkets operationella semantik.
- I kurserna inom funktionell programmering lär studenterna sig metoder och språkkonstruktioner som hjälper dem att snabbt konstruera korrekta program. Studenterna examineras dels kontinuerligt i form av laborationsuppgifter och dels en tentamen där för godkänt krävs 4 av 5 tillräckligt väl behandlade problem.
- I laborationerna i [“Programming Language Technology \(PLT\)”](#) och [“Compiler Construction”](#) går studenterna hela vägen från ett programspråks syntaktiska struktur via parsning, typsystem, transformationer och översättning till exekverbar kod. Där ingår också resursoptimering och utökningar av språk. Detta innebär att studenterna får djup metodkunskap om implementation av programspråk och erfarenheter av kopplingen mellan teori och praktik.

Delmål 2b: väsentligt fördjupade kunskaper inom vissa delar av det valda teknikområdet

En fundamental förutsättning för att detta delmål ska kunna uppfyllas är att en stor andel av lärarna är aktiva forskare, och att en majoritet av övriga har forskningserfarenhet, t ex i form av forskarutbildning. Se avsnittet om Lärarkapacitet och lärarkompetens i del 3 för mer information om detta. Chalmers civilingenjörsutbildning är uppbyggd av tre år på kandidatnivå och två års fördjupning på masternivå inom internationella masterprogram. D-studenterna fördelar sig i praktiken på fem olika masterprogram: CS-ALL, CSN, EESD, SE och ID&T och inom alla dessa uppnår man mycket väsentligt fördjupade kunskaper inom olika delar av teknikområdet. En generell kommentar är att många specialämnen bara kan behandlas översiktligt i de obligatoriska kurserna, men med hänvisningar till självständiga fördjupningar. Studenterna med specialintressen väljer sedan valfria kurser eller ett helt masterprogram som ger den önskade fördjupningen. Här beskriver vi kort de fem programmen.

Computer Science – Algorithms, Languages and Logic (CS-ALL)

Målet med CS-ALL är att ge en bred grund i datavetenskap som forskningsämne inom fyra huvudspår: **Algoritmer, Logik och verifisering, Programspråk** samt **Datasäkerhet**. Studenterna måste ta tre obligatoriska kurser: “Algorithms”, “Logic in Computer Science” och “Programming Language Technology” och minst fem profilkurser. Spåret Datasäkerhet är gemensamt med CSN-programmet och beskrivs under nästa delrubrik. Kurserna som bildar progressionen inom spåren listas nedan med andelen studenter som läser dem.

- **Algoritmer:** Algorithms (66 %), Artificial Intelligence (40 %), Advanced Algorithms (29 %), Algorithms for Machine Learning and Inference (21 %), Discrete Optimization (11 %). Inom detta spår finns forskningskoppling till såväl Bioinformatik som Business Intelligence (“big data”), företagskontakter med bl.a. Recorded Future och Jeppesen.
- **Logik och verifisering:** Logic in Computer Science (51 %), Software Engineering using Formal Methods (25 %), Models of Computation (14 %), Types for Programs and Proofs (12 %), Hardware Description and Verification (4 %). I bakgrunden finns mycket framgångsrik forskning inom formella metoder, programmeringslogik, typteori och testning. Exempel: EU-projekten [ForMath](#) (Thierry Coquand), och [ProTest](#) (John Hughes) samt [KeY-projektet](#) (Wolfgang Ahrendt).
- **Programspråk:** Functional programming (74 %), Programming Language Technology (68 %), Programming paradigms (44 %), Compiler Construction (29 %), Advanced Functional Programming (15 %), Frontiers of Programming Language Technology (6 %). Här får studenterna kontakt med en världsledande forskningsmiljö inom programspråk, funktionell programmering och språkteknologi. Industrirellevanta gemensamma projekt finns med Ericsson, Quviq och Statens Provningsanstalt. Exempel: EU-projekten [MOLTO – Multi-lingual Online Translation](#) (Aarne Ranta), [PROWESS - PROperty-based testing for Web Services](#) (John Hughes), [GSDP - Global Systems Dynamics and Policy](#) (Patrik Jansson).

Computer Systems and Networks (CSN)

CSN-programmets mål är att ge studenterna en gedigen förståelse för datorsystem och datornätverk genom att ge dem både bredd och djup inom detta viktiga område. Studenter som får en examen från CSN skall kunna (1) designa ett system baserat på nya och existerande komponenter (systemteknik), (2) Förstå interaktionen mellan hårdvara och mjukvara på låg nivå (datorarkitektur), (3) ha förmågan att utveckla systemet (programvaruteknik), och (4) analysera prestanda och begränsningar i det designade systemet (distribuerade system).

Programmets fyra obligatoriska kurser är “Fault-tolerant computer systems”, “Computer networks”, “Operativsystem” (som många studenter läst på kandidatnivån), samt i andra året “Advanced topics in CSN” för att knyta samman de olika spåren och förbereda för examensarbetet. Profilkurserna finns inom följande spår, där det första är gemensamt med CS-ALL:

- **Datasäkerhet:** Computer Security (62 %), Network Security (56 %), Cryptography (40 %), Programming Language Based Security (34 %). I detta spår² drar studenterna nytta av forskargruppens starka kopplingar till både akademi och industri. Exempel: “Open Web Application Security Project (OWASP)”, “Security Arena” på Lindholmen Science Park och “Urban Safety and Societal Security Research Center, URBSEC” samt världsledande forskning inom flera EU-projekt (Crisalis, WebSand, SysSec).
- **Parallella och distribuerade system:** Concurrent programming (77 %), Distributed systems (53 %), Distributed systems, advanced course (28 %), Parallel and distributed real-time systems (18 %), Parallel Functional Programming (ny), Parallel Computer Organization and Design (5 %). I detta spår lär sig studenterna de färdigheter som behövs för att hantera de senaste problemlösningsteknikerna och de storskaliga beräkningar som behövs för framsteg inom

algoritmer, system och programmering. Lärarkollegiet har starka kopplingar till flera lokala företag (bl.a. Ericsson och Nema Labs) och flera stora forskningsanslag (både nationellt och från EU).

- **Nätverk och kommunikation:** Internet technology (47 %), Wireless networks (39 %), Router och switchteknik (18 %), Multimedia and video communications (16 %). I detta spår lär sig studenterna analysera och utforma kommunikationsprotokoll, algoritmer, programvara och hårdvara som lämpar sig för datornätverk och kommunikation.
- **Datorsystemteknik:** Fault-tolerant computer systems (43 %), Real time systems (39 %), Computer architecture (12 %), Parallel Computer Organization and Design (5 %). I detta spår lär sig studenterna utforma system av program- och hårdvara: tekniker för att kontrollera datorsystem med strikta tidskrav; separation och moduluppdelning med virtuellt minne och trådar; atomicitet och koordination av parallella transaktioner; återstart och pålitlighet.
- **Programmering av datorsystem:** Databaser (78 %), Computer Graphics (37 %), Operativsystem (27 %), Unix internals (22 %). Detta spår handlar om design och programvaruutveckling för datorsystem som styr och kontrollerar datorns hårdvara och som erbjuder en plattform för att köra applikationsprogram.

Embedded Electronic System Design (EESD)

Designen av masterprogrammet [Embedded Electronic System Design](#) är influerad av det sk [CDIO-konceptet](#) och syftar till att "... educate engineers that can: conceive complex embedded system specifications; design such systems at the electronic system level (ESL); implement and verify such systems using state-of-the-art integrated circuit and packaging technologies and electronic design automation tools. The program provides a deep coverage of the methodologies critical to this conceive, design, implement and verify process. EESD graduates will be qualified to work as productive engineers in industrial teams designing and building state-of-the-art electronics products, and to undertake graduate studies leading to a doctorate in the field of electronic system design and packaging."

Redan de obligatoriska EESD-kurserna ger en fördjupning inom D-området [Introduction to electronic system design](#), [Introduction to integrated system design](#), [Methods for electronic system design and verification](#), [Embedded system design project \(15hp\)](#) och studenterna måste också välja tre till profilkurser.

Software Engineering (SE)

SE-programmet följer IEEE:s definition av Software Engineering som "(1) The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software, that is, the application of engineering to software, and (2) the study of the approaches as in (1)". Definitionen innebär att mjukvaruutveckling innefattar såväl processer för utveckling av mjukvaruprodukter som för förvaltning och underhåll av dessa. Tillämpning av systematiska metoder för mjukvaruutveckling innebär att utvecklingsprocesser är av särskild vikt för disciplinen då det understryks att mjukvaruutveckling är en ingenjördisciplin med dess karakteristiska stringens för planering, exekvering och uppföljning.

Programmet består av 7 obligatoriska ämneskurser som ger en mycket god fördjupning inom huvudområdet (60 hp), valbara kurser (30hp) samt examensarbete (30 hp). De obligatoriska kurserna är: Requirements Engineering, Software Project Management, Empirical Software Engineering, Model-Driven Engineering, Advanced Software Architecture, Agile Development Processes, samt Software Evolution Project (15 hp).

Interaction Design and Technologies (ID&T)

Interaktionsdesign handlar om design av interaktionen mellan människor och produkter i vilka informationsteknologi är en central komponent. Detta kan t ex vara design av nyttoprogram som körs på vanliga datorer, men också interaktionen mellan föraren och det komplexa system av datorer som styr en modern bil, design av moderna datorspel, s k appar för mobila enheter eller integration av IT i vardagssaker som ”intelligenta kläder”.

Avdelningen för interaktionsdesign vid Institutionen för Tillämpad Informationsteknologi ligger bakom masterutbildningen i interaktionsdesign. Vid avdelningen bedrivs forskning inom ett flertal områden såsom gränssnitt för mobila enheter, experimentella fysiska gränssnitt, speldesign, digital gestaltning, samt interaktionsdesign riktad mot fordonsindustrin. Forskningen vid avdelningen influerar innehållet i programmets kurser bl a genom att programmet har ett spår med inriktning på spelutveckling och ett annat med inriktning mot s k Tangible interfaces.

Programmet är upplagt med ett obligatoriskt block om 30 hp och fyra alternativobligatoriska spår om vardera 30 hp. I enlighet med Chalmers riktlinjer för masterprogram är 30 hp helt valfria kurser. Det finns dock möjlighet att välja att läsa även dessa inom huvudområdets kursutbud. De obligatoriska kurserna är: Interaction design methodology, Prototyping in interaction design, Graphical interfaces, Interaction design project 1.

Examensarbete

Examinatorerna för de 17 utvalda exjobbssrapporterna bedömer att dessa representerar hög kvalitet och att 70 % av dem motsvarar mycket hög kvalitet när det gäller väsentligt fördjupade kunskaper inom vissa delar av teknikområdet.

Examensmål 3

Färdighet och förmåga:

För civilingenjörsexamen skall studenten visa förmåga att med helhetssyn kritiskt, självständigt och kreativt identifiera, formulera och hantera komplexa frågeställningar samt att delta i forsknings- och utvecklingsarbete och därigenom bidra till kunskapsutvecklingen

Examensarbeten Civilingenjörsutbildningen innehåller två större självständiga arbeten. Först kandidatarbetet (15 hp) vilket gäller som examensarbete gentemot kandidatexamen och sedan masterexamensarbetet (30 hp²) vilket gäller som examensarbete för masternivå och gentemot civilingenjörsexamen. Övergripande Chalmersgemensam struktur och examinationskrav avseende båda dessa arbeten beskrivs i Del 3 (sid 30).

För civilingenjörsexamen kan kandidatarbetet ses som en obligatorisk projektkurs, och den betygsätts också individuellt som en normal kurs på skalan U, 3, 4, 5. I kandidatarbetet får teknologerna omfattande träning som syftar mot både examensmål 2 och 3, på grund av de komplexa och forsknings- eller utvecklingsanknutna frågeställningar som behandlas. I kandidatarbetet skall studenten integrera, fördjupa och utveckla sina kunskaper och färdigheter inom ett begränsat område av det som behandlats i tidigare kurser. Kandidatarbetet syftar också till att studenterna ska tillägna sig kunskaper och färdigheter i ingenjörsmässigt och vetenskapligt arbetssätt.

Kandidatarbetena utförs normalt på en institution på Chalmers och i grupper om 3–6 teknologer (median 4 för D-programmet). Detta medför att det blir svårare att uppnå och bedöma självständigheten, men det möjliggör samtidigt att betydligt mer komplexa och omfattande frågeställningar kan bli föremål för studien. Varje student måste agera självständigt för att uppfylla sin del av gruppens arbete, och flera examinationsmoment bedöms individuellt. Dessutom skall gruppen föra dagbok där individernas insatser journalförs. Muntlig presentation av projektet sker i grupper om max 3 studenter. Oppositionen sker individuellt, dels skriftligt före inlämning av slutversion av rapporterna, dels muntligt vid själva presentationen

Vi kan illustrera hur detta uppfattas med en studentkommentar från höstens kursutvärdering för Kandidatarbetskursen. På frågan *“Vad bör främst bevaras till nästa år?”* fick vi svaret *“studenternas egna ansvar, då det är detta jag tycker är det viktigaste med kandidatarbetet: att få göra ett projekt över en längre tid där man själv har ansvaret för vad som skall göras när”*.

Inom det avslutande masterexamensarbetet, vilket utförs inom en forskargrupp eller på ett företag, visar studenten att han/hon inte bara har samlat faktakunskaper, utan också att han/hon kan tillämpa och vidareutveckla denna kunskap med den nivå av självständighet som krävs för yrkesverksamhet som civilingenjör, eller för forskarstudier. Det är alltså en mycket konkret målsättning att studenten under masterexamensarbetet ska *“delta i forsknings- och utvecklingsverksamhet och därigenom bidra till kunskapsutvecklingen”*.

² I undantagsfall genomförs ett forskningsinriktat examensarbete om 60hp.

Delmål 3a: Förmåga att med helhetssyn kritiskt, självständigt och kreativt identifiera, formulera och hantera komplexa frågeställningar och därigenom bidra till kunskapsutvecklingen

Halvvägs i utbildningen, i det obligatoriska kandidatarbetet får studenterna öva och demonstrera sin förmåga att hantera komplexa frågeställningar. Men även innan dess finns en progression från mer styrda grundkurser till mer öppna frågeställningar i andra och tredje året.

På masternivån övas planering och självständig problemlösning i examensarbetet (de flesta i samarbete med industrin), i kursprojekt och i komplexa labuppgifter i exempelvis "Programming Language Technology (PLT)", "Artificial Intelligence" och "Language-based Security".

I PLT-kursen har den första laborationen lite av "hitta vägen i djungeln"-karaktär med en komplex och verklighetsnära implementationsuppgift. Vi beskrev mer om kursen på sida 12 och vi återkommer till den PLT-inspirerade läroboken senare (sida 19). Forsättningskursen "Frontiers of Programming Language Technology" krävs att studenterna kritiskt utvärderar nya programspråk och verktyg, effektivt tillgodogör sig och sammanfattar avancerade forskningsresultat samt gör medvetna val om utformning och användning av avancerad programspråksteknologi. Detta examineras genom seminarier, övningar och en projektrapport.

Inom kursprojekten i "Artificial Intelligence" måste studenterna integrera flera olika områden: språkbehandling, maskininlärning, sannolikhetslära och grafteori, planering och optimering.

Kursen "[Algorithms for Machine Learning and Inference](#)" handlar om att automatiskt förutsäga händelser och att klassificera komplexa objekt eller situationer baserat på ett underlag av data. För att utveckla (eller bara välja ut och använda) en meningsfull metod för ett problem av den här typen måste studenten kritiskt reflektera över vilka antaganden som används för modellen.

I algoritmkurser lär sig studenterna om olika paradigmer (exempelvis exakta eller approximativa lösningar, generella eller specialiserade algoritmer). Studenterna lär sig att hitta en kompromiss mellan generella lösningsmetoder å ena sidan och lösningens träffsäkerhet å andra sidan. Man måste välja en lämplig lösningsmetod för det aktuella problemet. Detta kräver en hel del kritisk och systematisk kunskapsintegration och examineras dels genom skriftliga svar på inlämningsuppgifter och dels på tentan.

I både den första algoritmkursen och i fortsättningskursen lär sig studenterna från föreläsningar, övningar och inlämningsuppgifter:

- hur man känner igen och formaliserar beräkningsproblem,
- hur man motiverar modelleringsbeslut (vad kan abstraheras bort och vad ingår i problemets kärna),
- hur man analyserar problemets struktur och hittar en algoritmisk metod,
- hur man omvandlar detta till en fullt specificerad algoritm och bevisar dess korrekthet,
- hur man analyserar dess effektivitet och ev. förbättrar den (ofta genom att undvika omräkning),
- hur man hittar alternativa metoder med andra föredelar och
- (om det behövs) hur man upprepar den här utvecklingsprocessen.

Det här är "lösningsmallen" för nästan alla problem som dessa kurser tar upp. Det finns inlämningsuppgifter som behandlar flera faser i den här mallen, fast ingen enskild uppgift täcker i sig alla faserna.

På det här sättet lär sig studenterna gemensamma designprinciper som kan användas på många liknande problem som kan dyka upp i praktiken. Begreppet "reduktion" (av ett problem till ett annat) formaliserar intuitionen att vissa problem är närbesläktade på så sätt att de alla kan lösas så snart en algoritm för ett av problemen är känd. Ett talande exempel är algoritmer för nätverksflöden som löser problem inom exempelvis transport, planering, schemaläggning, bildbehandling och informationsutvinning (data mining) trots att dessa problem inte uppenbart är närbesläktade med nätverksflöden. Genom återkommande träning bygger studenterna upp sin intuition och lär sig se sådana samband. Examinationen visar att förmågan att lösa komplexa problem mognar.

Delmål 3b: Förmåga att delta i forsknings- och utvecklingsarbete och därigenom bidra till kunskapsutvecklingen

Inom de två åren på avancerad nivå (för civ.ing.examen och masterexamen) är kurserna i allmänhet närmare F&U och de avslutas med examensarbetet som kronan på verket. Vi har redan i Delmål 1b (sida 9) sett hur studenterna får insikt i FoU och vi beskriver här kort hur förmågan att delta i FoU utvecklas i ett par olika spår och kurser:

- **Datasäkerhet:** Inom säkerhetsspåret finns A. Sabelfelds kurs "[Language-based security](#)" som kombinerar praktik med aktuellt forskningsmaterial och T. Olovssons kurs "[Network Security](#)" som innehåller flera forskningsartiklar i sin kurslitteratur (exempelvis "[The Final Nail in WEP's Coffin](#)"). I detta spår drar studenterna nytta av forskargruppens starka kopplingar till både akademi och industri. Exempel: "Open Web Application Security Project (OWASP)", "Security Arena" på Lindholmen Science Park och "Urban Safety and Societal Security Research Center, URBSEC" samt världsledande forskning inom flera EU-projekt (Crisalis, WebSand, SysSec). Detta spår har nyligen tagit fram en trevlig webbplats som beskriver kopplingen mellan undervisning, F&U och industri inom området: <http://www.cse.chalmers.se/edu/master/secspec/>
- **Parallella och distribuerade system:** I detta spår lär sig studenterna de färdigheter som behövs för att hantera de senaste problemlösningsteknikerna och de storskaliga beräkningar som behövs för framsteg inom algoritmer, system och programmering. Lärarkollegiet har starka kopplingar till flera lokala företag (bl.a. Ericsson och Nema Labs) och flera stora forskningsanslag (både nationellt och från EU). Två nya forskningsnära kurser har utvecklats under det senaste läsåret: Parallell Functional Programming, och Masterclass in Areas of Advance. Båda dessa har som målgrupp både avancerade masterstudenter och doktorander.
- **Types for programs and proofs:** Ett av examinationsmomenten i denna kurs är att muntligt presentera antingen en klassisk forskningsartikel, en mer nutida forskningsartikel eller ett av de mer avancerade kapitlen i [kursboken](#). Exempel på artiklar som presenterats är klassikerna J. Reynolds, [Towards a theory of type structure](#), P. Landin, [On the Mechanical evaluation of Expressions](#) och A. Sabelfeld³ and A. Myers, [Language-Based Information-Flow Security](#).
- **Advanced functional programming:** Varje år ingår det ett par gästföreläsningar inom aktuell forskning och det innebär att kursens innehåll varierar lite från år till år. Under 2012 var gästföreläsningarna om det domänspecifika språket "Feldspar" (för digital signalbehandling) som utvecklas i samarbete med Ericsson och om språket "Agda" (för bevis och program med starka invarianter) som utvecklas av programmeringslogikgruppen på Chalmers och GU. Studenternas betyg baseras till 60 % på programmeringsprojekt som anknyter till aktuell F&U.

Studenterna får möjlighet att delta i aktuell forskning i en del av de avancerade kurserna som exempelvis "Frontiers of programming language technology", "Advanced topics in computer systems

³ Andrei Sabelfeld är forskare och lärare inom Datasäkerhetsspåret på Chalmers. Artikeln har citerats 1193 gånger sedan den publicerades 2003.

and networks”, “Advanced functional programming” och “Language-based security”. Andra konkreta exempel är den nya kursen “Parallel functional programming” (utvecklad av John Hughes och Mary Sheeran) som bygger på aktuell forskning och beprövad erfarenhet inom området och som till stor del har vetenskapliga artiklar som källmaterial samt kursen “Algorithms for machine learning and inference” som har moderniserats under 2011/12 i samarbete med en gästforskare.

- Vi har utvecklat två nya kurser inom CSN-programmet under den gemensamma rubriken “Masterclass – Areas of Advance”. Den ena utvecklades i samarbete med Chalmers styrkeområde Transport och heter “Autonomous and Cooperative Vehicular Systems (LP2)”. Den andra utvecklas i samarbete med styrkeområdet Energi och heter “ICT Support for Adaptiveness and Security in the Smart Grid (LP4)”. Den gemensamma idén är forskningsnära handledning i en blandad grupp med studenter från mer än ett huvudområde och lärare från än en forskargrupp. Examination är dels i form av obligatoriska seminarier och projektrapporter.
- Kursen ”[Language-based security](#)” kombinerar praktiska datasäkerhetsfärdigheter med aktuellt forskningsmaterial. Den matematiska eller datalogiska grunden är användningen av formella och semantiska metoder för att modellera programs säkerhetsgenskaper. Flera av lärarna på kursen är involverade i Open Web Application Security Project (OWASP) och ett par av föreläsningarna var samtidigt föredrag inom det projektet. Studenterna uppmanas att delta i datasäkerhetstävlingar och i examinationen ingår ett projekt som redovisas både muntligt och skriftligt.
- Aarne Rantas föreläsninganteckningar i PLT har utvecklats under många år och har nu förädlats till en lärobok: [Implementing Programming Languages](#). Flera utkast har lästs och kommenterats av D-studenter och värdefulla förslag och idéer från studenterna har inkorporerats i den färdiga boken.
- Många studenter har i sina examensarbeten använt sina kunskaper inom programspråk genom att utforma och implementera nya domän-specifika språk på uppdrag av företag eller forskargrupper. Prof. Aarne Ranta har examinerat minst fem sådana projekt de senaste åren och hans intryck är att våra studenter allmänt anses ha goda kunskaper i implementation av programspråk. Många, men inte alla, dessa studenter tog också “Compiler Construction” för att ytterligare utveckla sina kunskaper och förmågor.

Studenterna har också möjligheten att ta en projektkurs och arbeta med ett fritt valt ämne (med coaching av en handledare). I väldigt lyckade fall kan det till och med leda till vetenskaplig publicering. Ett imponerande sådant exempel är följande artikel som skrevs när författaren var student på CS-ALL-programmet 2010. Han löste ett öppet problem inom ett par veckor:

Bassel Manna. Cluster editing problem for points on the real line: A polynomial time algorithm. *Information Processing Letters* 110 (21), 961–965 (2010).

Några av de bästa studenterna varje år har kapacitet att prestera på den här nivån, men publicering beror också på tur till viss del (rätt problem och rätt idé vid rätt tidpunkt).

Följande kurser från SE och ID&T visar hur studenter övar sin förmåga att delta i aktuell FoU:

- I kursen *Requirements Engineering* analyserar studenterna artiklar som beskriver aktuell forskning inom området. Studenterna tillämpar sina kunskaper från dessa artiklar i uppgifter där de visar hur de beskrivna metoderna kan användas i miniprojekt. För att godkännas på kursen måste studenterna också visa att de förstår relevanta standarder, t ex från IEEE.
- I kursen *Software Project Management* krävs att studenterna studerar minst två fallstudier hämtade från *Harvard Business Review*. Dessa studier är välkända inom området projektledning, där de används för att analysera multipla fenomen i ett fall ur flera skilda perspektiv. Genom analyserna tillägnar sig studenterna aktuell kunskap om metoder för projektledning och får värdera metoderna i relation till verkliga fall. I årets kurs valdes rapporten “Managing Knowledge and Learning at NASA and the Jet Propulsion Laboratory (JPL)” för närmare analys. För att genomföra analysen måste studenterna förstå principer för

projektledning och visa förmåga att analysera ett fall baserat på dessa principer.

- I kursen *Empirical Software Engineering* analyserar studenterna artiklar om forskningsmetodik i software engineering. Totalt analyseras 16 artiklar, varefter studenterna tillämpar sina vunna kunskaper i uppgifter där de beskriver hur empiriska studier kan utformas, till exempel hur en fallstudie utformas.
- I kursen *Agile Software Development* analyserar studenterna artiklar som beskriver aktuella utmaningar inom agil systemutveckling och erfarenheter från lokalt näringsliv om hur dessa metoder tillämpas. Kunskaperna från dessa analyser examineras i en skriftlig tentamen, där studenterna ska kunna reflektera över och jämföra olika typer av processer (exempelvis agila jämfört med plandrivna).
- Kursen *Interaction design methodology* har som kurslitteratur 17 uppsatser som antingen redovisar forskningsresultat eller är kapitel från central litteratur i ämnet. Litteraturen examineras i en hemtentamen. Studenterna uppmuntras också att själva söka fler källor för att stödja sin argumentation vid tentamen.
- I kursen *Human-centered design* används forskningsartiklar som kurslitteratur. Studenterna presenterar artiklar för varandra och måste även reflektera över innehållet genom att ta fram frågeställningar om de artiklar de inte presenterar själva. Studenternas förståelse för innehållet examineras i en serie hemtentamina där varje tentamen kräver att studenterna kan redogöra för och reflektera kring artiklarnas innehåll.

Examensmål 4

Färdighet och förmåga:

För civilingenjörsexamen skall studenten visa förmåga att utveckla och utforma produkter, processer och system med hänsyn till människors förutsättningar och behov och samhällets mål för ekonomiskt, socialt och ekologiskt hållbar utveckling

Delmål 4a: Förmåga att utveckla och utforma produkter, processer och system

När Chalmers ingenjörer från alla ämnen själva fick skatta sin ”förmåga att utveckla produkter, processer och system” i alumnenkäten 2012 angav de i snitt betyg 7 av max 10 vilket redan det är bra. D-ingenjörerna var ännu nöjdare och angav betyg 8 i snitt. Vår tolkning är att D-studenterna övar upp en mycket god sådan förmåga under studietiden och vi försöker här nedan beskriva några av de sätt detta syns i utbildningen.

Redan från första kursen i första året får D-studenterna börja öva sig på att utforma lösningar på labuppgifter. Detta börjar med mycket blygsamma ”produkter” men utvecklas snart till mer och mer krävande uppgifter i år två och de första större exemplen kommer i kandidatarbetsprojekten i år 3. På masternivån fortsätter progressionen och flera av programmen (SE, ID&T, EESD) har 15hp-projekt i fjärde året. De flesta studenterna utformar produkter, processer eller system i sina examensarbeten (30hp) på masternivå. Enligt vår bedömning gäller detta alla de utvalda 17 examensarbetena för civ.ing.examen från D. (Examensarbetena har behandlats tidigare under Examensmål 3, sid 16 och beskrivs i mer detalj i Del 3, sid 30).

Kandidatarbetsprojekten är för många studenter de mest verklighetsnära uppgifterna de genomför under sin utbildning. Istället för de vanliga labuppgifterna som ofta löses av två personer (som ofta känner varandra väl) under två veckor, möter studenterna här en större grupp (4-6 deltagare som de inte valt själva) och en längre tidsskala: 20 veckor.

Ett spännande stick-prov är kursen “Language-Based Security” som nämner bland sina mål “apply practical knowledge of security for modern programming languages. This includes the ability to identify application- and language-level security threats, design and argue for application- and language-level security policies, and design and argue for the security, clarity, usability, and efficiency of solutions, as well as implement such solutions in expressive programming languages.”

Programmeringsuppgifterna (data-race, överfulla buffrar, SQL-injektion och “cross-site scripting”) är handfasta och involverar attacker på och försvar av verkliga system. Övningarna är i form av webb-utmaningar. En stor del av kursen är projektarbete så att studenterna tränas på att genomföra självständiga undersökningar, ibland utan något enkelt svar. Sammantaget ger detta en mycket god förmåga att utveckla och utforma säkra datorsystem.

Delmål 4b: Förmåga att ta hänsyn till människors förutsättningar och behov och samhällets mål för ekonomiskt, socialt och ekologiskt hållbar utveckling

I [Chalmers lokala examensordning](#) finns två centralt beslutade obligatorier om Miljö och Hållbar Utveckling (MHU) och Människa, Teknik och Samhälle (MTS). För civ.ing. examen krävs både 7,5 hp inom MHU och 7,5 hp inom MTS. Samma kurs inte kan tillgodoräknas som fullgörande av båda obligatorierna för MHU och MTS, dvs varje student ska läsa totalt 15hp inom MHU och MTS.

För studenterna inom D-området är det framför allt den obligatoriska kursen i Bärkraftig resursanvändning (7,5 hp) som klassas som MHU. Kursen har följande lärandemål:

- redogöra för betydelsen av hållbar utveckling i dess tre huvudsakliga dimensioner: den ekologiska, den ekonomiska och den sociala
- förstå centrala analytiska koncept, (samt ha förmåga att använda dessa för att analysera hållbar utveckling) översiktligt redogöra för människans inverkan och effekter på jordens klimat- och ekosystem
- förstå begränsningar och möjligheter från hållbarhetssynpunkt för användningen av olika naturresurser och teknologier
- redogöra för utmaningar, strategier och styrmedel för hållbar resursanvändning
- förstå teknisk funktion hos olika nyckelteknologier och system för en hållbar resursanvändning
- förstå datorteknikens relevans som del i teknik och system för hållbar resursanvändning
- redogöra för olika möjligheter att från hållbarhetssynpunkt förbättra produkter och industriella processer inom data- och elektroniksektorerna
- (förstå centrala analytiska koncept,) samt ha förmåga att använda dessa för att analysera hållbar utveckling

Dessa lärandemål examineras genom två inlämningsuppgifter, fyra fördjupningsövningar, en dugga och en sluttentamen. Vi återkommer till den här MHU-kursen under Delmål 5 (sid 23) och 6 (sid 26).

De mest populära MTS-kurserna är "Teknik och samhälle" och "Teknikhistoria" men studenterna sprider ut sig på ett tiotal kurser som dock alla bidrar till förmågan att göra bedömningar med hänsyn till människan, tekniken och samhället.

De två nya kurserna inom CSN-programmet under den gemensamma rubriken "Masterclass – Areas of Advance" som vi beskrev på sida 19. I båda fallen är hållbar utveckling en av drivkrafterna och studenterna behöver hitta lösningar som tar hänsyn till både människan, miljön och maskinen.

Inom EESD-programmet tar studenterna en kurs i "Energy aware computing" där just de hållbar resursanvändning (med fokus på energiförbrukning) är grundtemat.

Examensmål 5

Färdighet och förmåga:

För civilingenjörsexamen skall studenten visa förmåga att i såväl nationella som internationella sammanhang muntligt och skriftligt i dialog med olika grupper klart redogöra för och diskutera sina slutsatser och den kunskap och de argument som ligger till grund för dessa

Chalmers har en mycket god internationell miljö med masterutbildning helt på engelska. Både lärargruppen och studentgruppen är blandad och vi har en levande akademisk miljö med gästforskare, utbytesstudenter och gästföreläsare från industrin. Detta medför att studenterna övar på en hel del informell kommunikation och dialog med olika grupper även utanför undervisningen. Viktigt att notera är också att kommunikationen under utbildningen inte bara handlar om två språk (svenska och engelska) utan det finns också mycket att lära sig inom teknisk terminologi, jargong, kulturell kontext och kroppsspråk. En roll som civilingenjörsutbildningen har är att ”socialisera in” studenterna i rollen som ingenjör i ett nationellt och internationellt sammanhang.

Delmål 5a: Förmåga att muntligt klart redogöra för och diskutera sina slutsatser och den kunskap och de argument som ligger till grund för dessa

Supplemental Instruction (SI) på datateknikprogrammet handlar om att ge teknologer en extra möjlighet, utöver den traditionella undervisningen och de vanliga övningarna, att komma in i studierna i matematikkurserna under den första terminen. Kort sagt går SI:n ut på att lära sig att lära sig och att lära sig att lära sig tillsammans med andra.

Oftast innehåller SI-pass därför något moment om studieteknik - tekniker för anteckningar, repetition, förberedelse mm. - och gruppövningar där man tränar på uppgifter om ämnen som nyligen tagits upp på föreläsningarna. Exempel från kursen i diskret matematik är delbarhet, primtal, induktionsbevis och från linjär algebra ortogonal projektion och matrisinvertering. Målet med dessa övningar är inte bara att man ska lösa uppgifterna, utan även att man ska lära sig att diskutera sig fram till en lösning tillsammans med andra, och även kunna förklara för andra varför en lösning är riktig och hur man kommit fram till den.

SI:n på datateknikprogrammet har varit framgångsrik på flera sätt. Den är uppskattad av teknologerna och den prisas i utvärderingar som ett bra komplement till den övriga undervisningen. Det finns även en tydlig samvariation mellan deltagande i SI och bra examinationsresultat.

SI-verksamheten ligger i första årskursen men det finns också exempel på andra kurser med inslag av muntlig kommunikation i examinationen: i “Datastrukturer” (årskurs 2) samt “Transformer, signaler och system” (årskurs 3) ger muntliga presentationer av lösningar framme vid tavlan bonuspoäng på tentan.

Kandidatarbetskursen har två lärandemål om muntlig kommunikation:

- 6. avrapportera kandidatarbetets resultat och hur dessa överensstämmer med arbetets projekt- och effektmål i skriftlig och muntlig form
- 7. kritiskt granska, värdera och konstruktivt ifrågasätta ett annat kandidatarbete avseende frågeställning, genomförande och resultat

I examinationen ingår att varje student skall genomföra muntlig presentation och opposition. Presentationen skall vara väl strukturerad med god användning av visualiseringshjälpmedlen samt

anpassad efter målgruppen och tidsramen. Den ska belysa syfte, upplägg och resultat. För godkänt skall oppositionen fokusera på helheten, ta upp de viktigaste frågeställningarna samt på ett bra sätt kritiskt belysa och diskutera innehållet.

Studenterna måste dra slutsatser från tekniska resultat och presentera dem i sina examensarbetsprojekt, både muntligt och skriftligt. Många andra kurser har också inlämningsuppgifter, projektuppgifter eller diskussionsövningar.

I kursen "[Types for programs and proofs](#)" ingår muntliga presentationer i par där studenterna presenterar forskningsartiklar eller bokkapitel. På så sätt får studenterna både lära sig ämnets vetenskapliga grund och öva på muntlig framställning.

I exempelvis "[Frontiers of programming language technology](#)" ingår det i målen att studenterna "ger effektiva och tydliga presentationer av mycket teknisk programspråksforskning" och det examineras i form av en seminarierie under ett läsår där studenterna presenterar forskningsartiklar muntligt med efterföljande diskussion. I examinationen ingår också närvaro på andra studenters seminarier och om man missar något eller några kan det kompenseras genom deltagande i reguljära forskningsseminarier inom D&IT-institutionen.

Delmål 5b: Förmåga att skriftligt klart redogöra för och diskutera sina slutsatser och den kunskap och de argument som ligger till grund för dessa

Skriftlig kommunikation är oerhört centralt i D-programmet, men majoriteten av den kommunikationen är inte med andra människor utan med datorsystem. Övningen på att skriva ekvationer, algoritmer, och programkod i olika formella språk börjar dag ett på D-programmet. Den snabba återkopplingen från matematisk programvara, kompilatorer, simulatorer och testramverk i kombination med handledning, övningar och föreläsningar gör att D-studenterna snabbt blir oerhört flinka på att uttrycka sina önsknings exakt och att mycket precist identifiera de delar som behöver lösas för att komma vidare. Steget härifrån till naturligt språk, slutsatser och underliggande argument är inte så långt som man kan tro, i varje fall inte i tekniska rapporter. Förutom att de är läsbara för datorer utgör program och specifikationer utgör också en viktig del av kommunikationen med andra ingenjörer och forskare. En intressant detalj är också att D-programmets lärarkollegium innehåller en stor och välkänd grupp inom språkteknologi, som har nära kontakter med Göteborgs universitets datalingsvistikgrupp och institutionen för svenska språket. Det innebär att intresserade studenter alltid har möjlighet att kombinera sina tekniska kunskaper med språkforskning inom olika områden.

Förutom själva programmen skriver D-studenterna också dokumentation, specifikationer, användarfall och rapporter. Ett exempel är kursen i [Matematisk statistik och diskret matematik](#) där två av tre punkter i kursens syfte handlar om kommunikation:

- färdighet i att förstå och använda det matematiska språket.
- förmåga att kommunicera matematik.

Ett av examinationsmomenten är en skriftlig rapport som dokumenterar en statistisk undersökning inklusive avgränsning, metodbeskrivning, analys och slutsatser.

I Datorsystemteknikkursen är en del av examinationen en skriftlig redovisning på engelska av en inlämningsuppgift. Ett annat exempel är kursen i Algoritmer där korta och koncisa skriftliga inlämningar krävs, så studenterna får öva sig på att formulera sig effektivt. I kursen Computer Security ingår skriftlig (och muntlig) examination av labuppgifter.

I Kandidatarbetskursen i tredje året ingår avdelningen för Fackspråk i lärolaget och studenterna övar både muntligt och skriftligt med återkoppling innan slutrapport, poster-session och muntlig

redovisning inklusive opposition. Fackspråk är också med i utvecklingen av en ny obligatorisk projektkurs som kommer att ligga i år 2 på D-programmet från hösten 2013 för att ge studenterna fler möjligheter att utveckla sin muntliga och skriftliga kommunikationsförmåga.

Kursen "[Fault-Tolerant Computer Systems](#)" har en större laborationsuppgift som examineras med skriftlig rapport enligt följande: "The results shall be documented in a short technical report. Pretend that you and your lab partner work together with a group of engineers to develop a new brake-by-wire system, and that your supervisor has given you the task to compare the dependability of a centralized and a distributed system architecture. The report shall present your findings to your supervisor and colleagues, who are skilled hardware and software designers but not necessarily dependability experts." Att på detta sätt explicit ange målgruppen ger uppgiften mer realistisk examination och studenterna är mycket nöjda med kursen.

Slutligen finns examensarbetet på masternivå där studenterna övas i att självständigt planera, genomföra och dokumentera en store projektuppgift. Här har SE-programmet under flera år hållit föreläsningar om planeringsrapport och slutrapportskrivning tidigt under arbetet och under detta läsår sprider vi detta till hela D-området.

Delmål 5c: Förmåga till dialog med olika grupper

Kandidatarbetet utförs i grupper om 3-6 studenter och är Chalmersövergripande upplagt så att det kan utföras tillsammans av studenter från olika civilingenjörsprogram och från datavetenskapligt program på Göteborgs universitet. Inom civilingenjörsexamen D är detta vanlig förekommande - vi har sällan grupper med studenter bara från ett program. Studenterna tränas i detta sammanhang på att kommunicera med personer som har annan ämnesbakgrund och ämneskultur. Genom det obligatoriska responsarbetet vid handledningstillfälle och oppositionen vid slutredovisningen får studenterna övning i dialog genom att ge återkoppling på andra studentgruppers arbeten och ta emot sådan återkoppling på det egna arbetet. Många studentkommentarer från kursutvärderingen för kandidatarbetet lyfter kommunikationsdelen på frågan "Vad bör främst bevaras till nästa år". Ett exempel är: "kontakten med Fackspråk var mycket bra. Likaså bibliotekets kurser i källkritik och informationssökning."

Studenterna måste dra slutsatser från tekniska resultat och presentera dem i sina examensarbetsprojekt, både muntligt och skriftligt. Många andra kurser har också inlämningsuppgifter, projektuppgifter eller diskussionsövningar. I exempelvis "Frontiers of Programming Language Technology" ingår det i målen att studenterna "ger effektiva och tydliga presentationer av mycket teknisk programspråksforskning" och det examineras i form av en seminarierie på 1/4-fart under läsåret där studenterna presenterar forskningsartiklar.

Examensmål 6

Värderingsförmåga och förhållningssätt:

För civilingenjörsexamen skall studenten visa insikt i teknikens möjligheter och begränsningar, dess roll i samhället och människors ansvar för hur den används, inbegripet sociala och ekonomiska aspekter samt miljö- och arbetsmiljöaspekter

När Chalmers ingenjörer själva fick skatta sin värderingsförmåga enligt detta mål i alumnenkäten 2012 angav de i snitt betyg 7 av max 10 vilket redan det är bra. D-ingenjörerna var ännu nöjdare och angav betyg 8,5 i snitt. Det kanske beror på att det inom detta mål finns många aspekter som är relevanta från ett D-perspektiv: integritetsfrågor (Datakommunikation, Computer Security); möjliga bidrag till demokratiska processer (Internetteknik, Människa-dator interaktion); resurs- o miljöstyrning (Bärkraftig resursanvändning, Algorithms, Optimization).

I avsnittet om Delmål 4b (sid 21) beskrev vi hur D-studenterna enligt den lokala examensordningen måste läsa en MHU och en MTS-kurs. Dessa kurser ger också studenterna insikt om teknikens möjligheter och begränsningar.

En icke-teknisk aspekt på maskininlärning är att resultaten av självlärande program alltid baseras på (och är starkt beroende av) modellantaganden - så kallad "inductive bias". Det är därför alltid användarna som skall bestämma om tolkning och användning av resultat, exempelvis för beslutsfattande. Dessa modellantaganden är en viktig punkt i alla delar av kursen.

Den (bevisbara) svår Lösbarheten av många beräkningsproblem samt oavgörbarhet (för exempelvis andra ordningens predikatlogik) kan ses som en inneboende begränsning hos datatekniken och denna aspekt uppmärksammas både i kärnkurser och i vissa profilkurser (exempelvis "Models of Computation"). Genom att lära sig begreppet NP-komplett (vilket också examineras) blir studenter medvetna om vissa problems svår Lösbarhet, dvs. de blir medvetna om att det finns djupa principiella skäl att inte alla formaliserade problem kan lösas med datateknik. De lär sig också sätt att hantera sådana svåra problem i praktiken i form av approximativa algoritmer för problem som inte kan lösas exakt. Studenterna lär sig också en intressant vinkling på detta i den populära kryptoteknik-kursen, där just svårigheten att invertera vissa beräkningar utnyttjas till säker kodning av känsliga data.

I Chalmers nya "[Riktlinjer för betygssättning av exjobb](#)" finns det lärandemål som berör detta delmål (men de har ännu inte hunnit användas för de som examinerades förra läsåret):

- *Inom ramen för det specifika examensarbetet kunna identifiera vilka frågeställningar som behöver besvaras för att hållbar utveckling skall beaktas:*
 - G Redovisar och motiverar valda metoder och diskuterar resultat utifrån ett livscykelperspektiv med fokus på hållbar utveckling.
 - IG Beaktar inte denna aspekt, trots att den av examinator bedöms vara av betydelse för det aktuella examensarbetet.
- *Medvetenhet om etiska aspekter på forsknings- och utvecklingsarbete:*
 - G Redovisar möjliga etiska konsekvenser av utfört arbete på forsknings- och utvecklingsarbete.
 - IG Beaktar inte denna aspekt, trots att den av examinator bedöms vara av betydelse för det aktuella examensarbetet.

Software Engineering.

Detta mål uppfylls genom bidrag från flera kurser, såsom *Requirements Engineering, Empirical*

Software Engineering, och *Software Evolution Project*. Programmet arbetar systematiskt med samhällliga och etiska aspekter i två spår. Det första spåret är etiska aspekter av mjukvaruutveckling och forskning i området. Det spåret beaktas, lärs ut och examineras huvudsakligen i två obligatoriska kurser: *Empirical Software Engineering* och *Software Project and Quality Management*. Det andra spåret är de samhällliga aspekterna av mjukvaruutveckling. Det spåret examineras huvudsakligen i den obligatoriska kursen *Software Evolution Project* där studenter skall reflektera över hur deras resultat påverkar förhållanden för deras ”kunder”, dvs. de företag som är mottagare av resultaten.

Kursen *Empirical Software Engineering* adresserar etiska aspekter av forskningsarbetet genom att studenter diskuterar, analyserar och reflekterar över generella etiska aspekter av forskning, t ex Nuremberg Code⁴ och i synnerhet etiska aspekter av datainsamling, planering och exekvering av empiriska studier. Studenter måste läsa två obligatoriska artiklar som diskuterar etiska aspekter på forskning i mjukvaruutveckling. För att bli godkänd på kursen krävs det att studenten reflekterar över de etiska aspekter som diskuteras i kursen. Även IEEE Code of Ethics⁵ diskuteras i kursen som en möjlig etisk standard inom området mjukvaruutveckling.

Kursen fokuserar på forskningsarbete på masternivå och i samhället i stort med avseende på forskningsmetodik, men även på forskningsetik. I kursen studeras och examineras hur etiska riktlinjer påverkar forskningsarbetet. Studenter diskuterar rollen av etikprovning, vilka krav som ställs för empiriska studier och hur etiska studier bör bedrivas.

I kursen diskuteras också begrepp som självständighet, originalitet, innovation och vad plagiat betyder samt dess konsekvenser. Dessa aspekter beaktas även vid handledningen av de självständiga arbetena

⁴ <http://history.nih.gov/research/downloads/nuremberg.pdf>

⁵ <http://www.ieee.org/about/corporate/governance/p7-8.html>

Avslutande slutsatser om måluppfyllelse – del 1

Examinatorerna för de utvalda exjobben är överlag nöjda med studenternas prestationer och många av dem är mycket nöjda. Studenterna är själva mycket nöjda: Högskoleverkets alumnienkät avslutas med frågan

F10: Om valet skulle ske idag, skulle du då påbörja samma utbildning?
och alla D-alumni svarade ”Ja” (svarsfrekvensen på enkäten var 69 %). Sammanfattningsvis hävdar vi att D-studenterna som får en civilingenjörsexamen från Chalmers väl uppfyller målen.

Del 2

Lärarkompetens och lärarkapacitet

Lärartabellen visar examinatorer för de kurser som tas av minst 10 % av de examinerade studenterna. Av dessa totalt 79 forskare och lärare har 91 % doktorsexamen och 67 % dessutom minst docent. Tre fjärdedelar delar sin tid mellan forskning och undervisning och en fjärdedel är heltidslärare (flera med ledaruppdrag). Utöver detta är ett stort antal doktorander aktiva som övningsledare och labassistenter, till en maximal omfattning av 20 % av heltid (per person).

För att bli anställd som eller befordrad till lektor, docent, biträdande professor eller professor krävs sedan mitten på nittioalet enligt Chalmers arbetsordning högskolepedagogiska meriter motsvarande 15 hp. Alla doktorander går en obligatorisk pedagogisk kurs om 3 hp.

Antal helårsstudenter

Antal helårsstudenter i aktuell utbildning

| | Antal |
|-----------------|-------|
| Helårsstudenter | 420 |

Civilingenjörsutbildningen i Datateknik omfattar **420 HST** fördelat på **240 HST** på kandidatnivå och **180 HST** på masternivå.

Analys och slutsats

Tillgången på kompetenta lärare är mycket god. Nästan alla föreläsare är aktiva forskare, som på både grundläggande och avancerad nivå undervisar inom områden som är länkade till forskningsinriktningen. Bredd och samhällskontakt är tydlig i form av olika gästföreläsare som undervisar inom sitt expertisområde.

Del 3

Andra förhållanden

Om examensarbetet

Chalmers tillämpar sedan 2009 en gemensam modell för examensarbete för civilingenjör- och teknologie masterexamen⁶. Modellen omfattar gemensamma regler för förkunskaper, för lärandemål, och för examination. Med den gemensamma modellen har lärandemålen förtydligats, informationen till studenterna förbättrats, studenttrörlighet inom Chalmers förenklats och ambitionsnivån höjts, speciellt vad gäller förmågan att planera och genomföra kvalificerade uppgifter på avsedd tid, och färdigheter i skriftlig och muntlig kommunikation på engelska.

Examensarbete vid Chalmers civilingenjör/arkitekt/masterprogram omfattar 30 eller 60 hp, och genomförs vanligen den sista terminen på programmen. Examensarbetet genomförs av 1-2 personer. När två studenter utför examensarbetet tillsammans skall arbetsfördelningen tydligt framgå av examensarbetsrapporten.

Förkunskaper

På Chalmers ställs höga krav för att få påbörja examensarbete, både vad gäller fördjupade ämneskunskaper och förmåga att hantera komplexa frågeställningar, och på muntlig och skriftlig kommunikationsförmåga. Dessa förkunskaper och -färdigheter etableras initialt i kandidatdelen av programmet, speciellt kandidatarbetet, och vidareutvecklas i masterdelens obligatoriska kurser och projekt. Chalmers har ett generellt krav på att minst 45 hp kurser inom masterprogrammet skall vara avklarade för att få påbörja examensarbete. Därutöver kan examinator kräva att specifika kurser avklarats som är nödvändiga för det enskilda examensarbetets genomförande.

Lärandemål

Chalmers lärandemål för examensarbeten på civilingenjör/arkitektutbildning är baserade på de mål som finns beskrivna i examensordningen. Ett medvetet urval har gjorts av de av examensordningens mål som kan visas i examensarbetet. Eftersom examensarbeten på civilingenjör/arkitektprogram är gemensamma med masterprogrammets så inkluderas även mål som är specifika för masterexamen i examensarbetenas lärandemål. Tabell X nedan redovisar Chalmers lärandemål för examensarbete och hur de relaterar till de mål som bedöms i föreliggande utvärdering. Samma formulerade lärandemål gäller för examensarbeten omfattande såväl 30 som 60 högskolepoäng (hp). Examensarbete omfattande 60 hp skall ha en väsentligt högre ambitionsnivå vad gäller vetenskaplig nivå eller teknisk/arkitektonisk realisering.

⁶ Chalmers tekniska högskola: Föreskrifter för examensarbete på civilingenjör- arkitekt- och masterprogram, Chalmers tekniska högskola, Göteborg, 2009.

Tabell X. Chalmers lärandemål för examensarbete på avancerad nivå vs de mål som utvärderas i 2012 års nationella utvärdering.

| Chalmers lärandemål för examensarbete för civilingenjör/arkitekt/masterexamen. Studenten skall demonstrera | Mål för civilingenjörsexamen | Mål för masterexamen |
|--|-------------------------------------|-----------------------------|
| väsentligt fördjupade kunskaper inom huvudområdet/inriktningen för utbildningen inkluderande fördjupad insikt i aktuellt forsknings- och utvecklingsarbete, | 1, 2 | 1a |
| fördjupad metodkunskap inom huvudområdet/inriktningen för utbildningen, | | 1b |
| förmåga att bidra till forsknings- och utvecklingsarbete, | 3 | |
| förmåga att med helhetssyn kritiskt, självständigt och kreativt identifiera, formulera och hantera komplexa frågeställningar, | 3 | 3 |
| förmåga att planera och med adekvata metoder genomföra kvalificerade uppgifter inom givna ramar, samt att utvärdera detta arbete, | | 3 |
| förmåga att skapa, analysera och kritiskt utvärdera olika tekniska/ arkitektoniska lösningar, | | |
| förmåga att kritiskt och systematiskt integrera kunskap | | 2 |
| förmåga att på engelska muntligt och skriftligt klart redogöra för och diskutera sina slutsatser, samt den kunskap och de argument som ligger till grund för dessa, | 5 | 4 |
| förmåga att inom ramen för det specifika examensarbetet kunna identifiera vilka frågeställningar som behöver besvaras för att alla relevanta dimensioner av hållbar utveckling skall beaktas, samt | 6 | |
| medvetenhet om etiska aspekter på forsknings- och utvecklingsarbete. | | 5 |

Genomförande

Studentens första steg i examensarbetet är att söka efter ett lämpligt projekt. Många studenter inom D-området gör sina examensarbeten i industrin och hittar förslag på företags hemsidor eller via gästföreläsare inom masterskurser. En annan stor grupp gör examensarbeten i en av Chalmers forskargrupper och får förslag den vägen. En projektbeskrivning tas fram som innehåller bakgrund, syfte, mål och eventuellt metod för det föreslagna arbetet. Examinator kontrollerar att studenten uppfyller de generella och specifika förkunskapskraven för examensarbete, och att det föreslagna

examensarbetet motsvarar lärandemålen för examensarbete. Examinator bedömer och ansvarar speciellt för att det föreslagna examensarbetet har en nivå motsvarande en väsentlig fördjupning inom området. För externa examensarbeten kontrolleras att en extern handledare finns identifierad och är införstådd med Chalmers regelverk för examensarbeten, och att nödvändiga resurser för examensarbetets genomförande i form av till exempel mjukvara, datorer eller experimentell utrustning finns tillgängliga. Masterprogramansvarig kontrollerar att examensarbetet ämnesmässigt faller inom masterprogrammets del av huvudområdet och ger klartecken för att starta arbetet.

Under de första veckorna av examensarbetet skriver studenten en planeringsrapport som preciserar problembeskrivningen/uppgiften. Den ska innehålla bakgrund, syfte, mål, avgränsningar, metod och tidsplan för examensarbetets genomförande. Examinator granskar och godkänner planeringsrapporten.

Under arbetets gång har studenten tillgång till regelbunden handledning (typiskt 1 h/vecka) från examinator eller av denne utsedd handledare vid Chalmers, samt, om examensarbetet görs utanför Chalmers av en extern handledare.Handledningsfrekvensen är ofta högre i början av arbetet då planeringsrapporten tas fram och i slutfasen då slutrapporten skrivs.

Examensarbetet avslutas med spikandet av slutrapporten och den muntliga presentationen och försvarandet av detta i ett öppet seminarium. Vid presentationen tränas förmåga till muntlig kommunikation i internationella sammanhang och förmågan att i dialog med forskare, yrkesverksamma ingenjörer och studenter försvara och förklara sina resultat. För 60 hp examensarbeten görs en särskilt rapportering av arbetets status till examinator och masterprogramansvarig.

Examination

Examensarbetena examineras med skriftliga och muntliga underlag. Såväl skriftlig som muntlig redovisning sker på engelska.

I det skriftliga underlaget ingår projektbeskrivning, planeringsrapport och examensarbetsrapport: Projektbeskrivningen och planeringsrapporten är en formativ examination och har beskrivits ovan. Den slutliga examensarbetsrapporten bedöms av examinator mot ett antal kriterier motsvarande lärandemålen enligt ovan. För varje lärandemål finns kriterier motsvarande bristande kvalitet, hög kvalitet och mycket hög⁷. Till exempel, krävs för Chalmers lärandemål *"Väsentligt fördjupade kunskaper inom huvudområdet/inriktningen för utbildningen inkluderande fördjupad insikt i aktuellt forsknings- och utvecklingsarbete"* följande för godkänt (hög kvalitet) *"En väsentlig fördjupning inom huvudområdet är demonstrerad. Arbetet utnyttjar kunskaper från studier på avancerad nivå inom huvudområdet. En skriftlig genomgång av befintlig litteratur samt att en reflektion över arbetets koppling till kunskapsfronten inom huvudområdet finns"*, men underkänt (bristande kvalitet) kan föranledas av *"Arbetets koppling till huvudområdet är svag eller saknas. Kunskaper från avancerad nivå inom huvudområdet utnyttjas inte. Litteratursammanställning samt reflektion över arbetets koppling till tillhörande kunskapsområde saknas"*. Brister vad gäller ett eller flera lärandemål i kombination kan leda till underkänt på hela examensarbetet. Kriterier för mycket hög kvalitet finns även formulerade men används främst för att ge återkoppling till studenterna; betyg ges i skalan underkänt-godkänt. För 60 hp examensarbete görs även en skriftlig rapportering i mitten av examensarbetet, i form som anvisas av examinator. Examensarbetet granskas även med hjälp av ett

⁷ Chalmers tekniska högskola: Riktlinjer för bedömning av kvalitet på examensarbete vid Chalmers civilingenjörskurs-/arkitekt- och masterprogram, Chalmers tekniska högskola, Göteborg, 2011.

plagiatverktyg, Urkund. Examensarbetsrapporten publiceras elektroniskt i Chalmers Publication Library.

Den muntliga redovisningen skall demonstrera att studenten har ”förmåga att på engelska muntligt (och skriftligt) klart redogöra för och diskutera sina slutsatser, samt den kunskap och de argument som ligger till grund för dessa”. De muntliga examinationsmomenten omfattar presentation och försvar av eget arbete, opposition på ett annat arbete, samt närvaro vid två andra examensarbetsredovisningar. Muntlig redovisning ska alltid genomföras på Chalmers men görs oftast även på det företag som examensarbetet gjorts vid.

Analys

Chalmers examensarbeten inom Datateknik har mycket stor relevans för studentens framtida yrkesverksamhet. Det finns ett stort intresse från näringslivet att erbjuda examensarbeten. 76 %⁸ av Chalmers civilingenjörstudenter gör examensarbeten inom industri eller offentlig verksamhet och för D-området ligger det något högre ändå. 39 % av Chalmers alumner nämner examensarbetet som en viktig faktor för att de fick sin första anställning, den tredje viktigaste faktorn efter utbildningens hela inriktning och Chalmersexamen som sådan. Alumnenkäten visar vidare att studenterna i hög grad utvecklat de kunskaper/färdigheter som exjobbet fokuserar på dvs. att de uppnått en väsentlig fördjupning.

Chalmers process med granskning av examensarbetsförslag av examinator och masterprogram-ansvariga säkerställer att de examensarbeten som får klartecken för att starta uppfyller våra och examensordningen krav på väsentlig fördjupning inom huvudområdet.

Slutsats

Examensarbeten inom Datateknik vid Chalmers bidrar starkt till måluppfyllelsen av utvärderingsmål 1-5. Examensarbetenas relevans för yrkesverksamhet evideras av den stora andel examensarbeten som görs i näringslivet och att de av alumni bedöms som en av de tre viktigaste faktorerna för att de fick sin första anställning.

Chalmers gemensamma process för examensarbeten säkerställer att examensarbeten adresserar en problemställning inom programmets huvudområde, innebär en väsentlig fördjupning av ämneskunskaper, och demonstrerar förmåga att självständigt arbeta med kvalificerade uppgifter, samt god förmåga till muntlig och skriftlig kommunikation i internationella sammanhang.

Denna rapport innehåller ca 75 000 tecken fram hit.

⁸ Chalmers tekniska högskola: Alumnenkät till examinerade 2009, Chalmers, Göteborg, 2012

Bilaga: tabell – Lärarkompetens och lärarkapacitet

| Namn | Titel | Huvudområden | Nivåer | Tjänstebenäm. | Forskn | GRU | Roll |
|--------------------|------------------|--|---------------------|----------------|--------|-----|--|
| Andrei Sabelfeld | Docent | Datateknik, Informationsteknik | Avancerad | Bitr professor | 80 | 20 | Examinator, kursansvarig |
| Devdatt Dubhashi | Bitr professor | Datateknik, Informationsteknik | Avancerad | Bitr professor | 43 | 57 | Examinator, kursansvarig |
| Erik Bohlin | Bitr professor | Industriell ekonomi | Grund | Bitr professor | 20 | 80 | Examinator, kursansvarig |
| Erland Jonsson | Bitr professor | Datateknik, Informationsteknik | Avancerad | Bitr professor | 85 | 15 | Examinator, kursansvarig |
| Jan Jonsson | Docent | Datateknik, Elektroteknik, Informationsteknik | Avancerad | Bitr professor | 10 | 20 | Examinator, kursansvarig |
| Kjell Jeppson | Bitr professor | Datateknik, Elektroteknik | Avancerad | Bitr professor | 50 | 50 | Masterprogramansvarig, examinator, kursansvarig |
| Koen Claessen | Docent | Datateknik, Informationsteknik | Grund | Bitr professor | 45 | 15 | Examinator, kursansvarig |
| Morten Fjeld | Docent | Datateknik, Informationsteknik | Grund | Bitr professor | 90 | 10 | Examinator, kursansvarig |
| Patrik Jansson | Docent | Datateknik, Informationsteknik | Avancerad | Bitr professor | 50 | 50 | Programansvarig D, examinator, kursansvarig |
| Serik Sagitov | Bitr professor | Matematik | Grund | Bitr professor | 65 | 35 | Examinator, kursansvarig |
| Åke Fäldt | Bitr professor | Teknisk fysik | Grund | Bitr professor | 0 | 73 | Examinator, kursansvarig |
| Ana Bove | Docent | Datateknik, Elektroteknik, Informationsteknik, Matematik | Grund, Avancerad | Docent | 45 | 20 | Studierektor, Examinator, kursansvarig |
| Ann-Sofie Axelsson | Docent | Industriell ekonomi | Grund | Docent | 20 | 70 | Examinator, kursansvarig |
| Dag Wedelin | Docent | Matematik | Grund | Docent | 30 | 30 | Examinator, kursansvarig |
| Elad Schiller | Forskarassistent | Datateknik, Informationsteknik | Avancerad | Docent | 30 | 70 | Masterprogramansvarig, examinator, kursansvarig |
| Fang Chen | Docent | Datateknik | Avancerad | Docent | 59 | 41 | Examinator, kursansvarig |

| Namn | Titel | Huvudområden | Nivåer | Tjänstebenäm. | Forskn | GRU | Roll |
|-------------------------|------------------|---|------------------|------------------|--------|-----|---|
| Graham Kemp | Docent | Biotechnik, Datateknik, Informationsteknik | Grund, Avancerad | Docent | 40 | 60 | Masterprogramansvarig, examinator, kursansvarig |
| Johan Karlsson | Docent | Matematik | Grund | Docent | 0 | 100 | Examinator, kursansvarig |
| K V S Prasad | Docent | Datateknik, Informationsteknik | Avancerad | Docent | 40 | 60 | Examinator, kursansvarig |
| Lars R Bengtsson | Docent | Datateknik, Elektroteknik | Grund | Docent | 25 | 25 | Examinator, kursansvarig |
| Lars Svensson | Docent | Datateknik | Avancerad | Docent | 25 | 50 | Examinator, kursansvarig |
| Marina Papatrifiantilou | Docent | Datateknik, Elektroteknik, Informationsteknik | Grund | Docent | 55 | 45 | Examinator, kursansvarig |
| Miroslaw Staron | Univ lektor | Datateknik, Elektroteknik, Informationsteknik | Grund, Avancerad | Docent | 50 | 50 | Viceprefekt för grundutbildningen, examinator, kursansvarig |
| Olof Torgersson | Univ lektor | Datateknik, Informationsteknik | Avancerad | Docent | 50 | 50 | Masterprogramansvarig, examinator, kursansvarig |
| Per Lundgren | Docent | Elektroteknik | Avancerad | Docent | 60 | 40 | Pedagogisk utvecklingsledare |
| Peter Damaschke | Docent | Datateknik, Informationsteknik | Avancerad | Docent | 45 | 55 | Examinator, kursansvarig |
| Robert Feldt | Univ lektor | Datateknik, Informationsteknik | Avancerad | Docent | 65 | 35 | Examinator, kursansvarig |
| Sally McKee | Docent | Datateknik | Avancerad | Docent | 75 | 25 | Examinator, kursansvarig |
| Staffan Björk | Docent | Datateknik, Informationsteknik | Avancerad | Docent | 50 | 50 | Examinator, kursansvarig |
| Stefan Lemurell | Docent | Matematik | Grund | Docent | 25 | 65 | Examinator, kursansvarig |
| Sven-Arne Andreasson | Docent | Datateknik, Elektroteknik, Informationsteknik | Grund, Avancerad | Docent | 0 | 100 | Examinator, kursansvarig |
| Ulf Assarsson | Docent | Datateknik, Informationsteknik | Avancerad | Docent | 50 | 50 | Examinator, kursansvarig |
| Alejandro Russo | Forskarassistent | Datateknik, Informationsteknik | Grund | Forskarassistent | 80 | 20 | Examinator, kursansvarig |
| Nils Anders Danielsson | Forskarassistent | Datateknik, Informationsteknik | Grund | Forskarassistent | 70 | 30 | Examinator, kursansvarig |

| Namn | Titel | Huvudområden | Nivåer | Tjänstebenäm. | Forsk | GRU | Roll |
|---------------------|----------------|--|---------------------|---------------|-------|-----|--------------------------|
| Aarne Ranta | Professor | Datateknik, Informationsteknik | Avancerad | Professor | 80 | 20 | Examinator, kursansvarig |
| Bengt Berglund | Professor | | Grund | Professor | 25 | 75 | Examinator, kursansvarig |
| Bengt Nordström | Professor | Datateknik, Informationsteknik | Avancerad | Professor | 5 | 30 | Examinator, kursansvarig |
| Bo Egardt | Professor | Elektroteknik | Grund | Professor | 61 | 29 | Examinator, kursansvarig |
| David Sands | Professor | Datateknik, Informationsteknik | Grund, Avancerad | Professor | 70 | 30 | Examinator, kursansvarig |
| Erik Ström | Professor | Datateknik | Avancerad | Professor | 65 | 20 | Examinator, kursansvarig |
| Irene Gu | Professor | Datateknik | Avancerad | Professor | 85 | 15 | Examinator, kursansvarig |
| Jan Smith | Professor | Datateknik, Informationsteknik | Avancerad | Professor | 50 | 10 | Examinator, kursansvarig |
| Johan Karlsson | Professor | Datateknik | Avancerad | Professor | 75 | 25 | Examinator, kursansvarig |
| John Hughes | Professor | Datateknik, Informationsteknik | Grund, Avancerad | Professor | 80 | 20 | Examinator, kursansvarig |
| Jörgen Hansson | Professor | Datateknik, Informationsteknik | Avancerad | Professor | 57 | 8 | Examinator, kursansvarig |
| MariAnne Karlsson | Bitr professor | Datateknik, Informationsteknik | Avancerad | Professor | 71 | 29 | Examinator, kursansvarig |
| Mary Sheeran | Professor | Datateknik, Elektroteknik, Informationsteknik | Avancerad | Professor | 80 | 20 | Examinator, kursansvarig |
| Olle Nerman | Professor | Datateknik | Avancerad | Professor | 65 | 35 | Examinator, kursansvarig |
| Per Larsson-Edefors | Professor | Datateknik, Elektroteknik | Grund, Avancerad | Professor | 60 | 40 | Examinator, kursansvarig |
| Per Stenström | Professor | Datateknik | Avancerad | Professor | 75 | 25 | Examinator, kursansvarig |
| Peter Dybjer | Professor | Datateknik, Informationsteknik | Avancerad | Professor | 60 | 15 | Examinator, kursansvarig |
| Philippas Tsigas | Professor | Datateknik | Avancerad | Professor | 80 | 20 | Examinator, kursansvarig |
| Reiner Hähnle | Professor | Datateknik, Informationsteknik | Grund, Avancerad | Professor | 70 | 30 | Examinator, kursansvarig |
| Thierry Coquand | Professor | Datateknik, Informationsteknik | Avancerad | Professor | 90 | 10 | Examinator, kursansvarig |

| Namn | Titel | Huvudområden | Nivåer | Tjänstebenäm. | Forskn | GRU | Roll |
|--------------------|--------------|---|------------------|---------------------|--------|-----|--|
| Tomas McKelvey | Professor | Automation och mekatronik, Datateknik, Elektroteknik, Kemiteknik, Teknisk fysik | Grund, Avancerad | Professor | 80 | 20 | Examinator, kursansvarig |
| Arne Linde | Tekn lic | Datateknik, Elektroteknik, Informationsteknik | Grund | Tekn lic | | 50 | Examinator, kursansvarig |
| Peter Lundin | Tekniklektor | Datateknik | Grund | Tekniklektor | | 55 | Examinator, kursansvarig |
| Rolf Snedsböl | Tekniklektor | Datateknik | Grund | Tekniklektor | | 75 | Studierektor, Examinator, kursansvarig |
| Sven Knutsson | Tekniklektor | Datateknik, Elektroteknik, Informationsteknik | Avancerad | Tekniklektor | 0 | 100 | Examinator, kursansvarig |
| Ali S Sahleson | Univ adjunkt | Datateknik, Elektroteknik, Informationsteknik | Grund | Universitetsadjunkt | 0 | 100 | Examinator, kursansvarig |
| Erland Holmström | Univ adjunkt | Datateknik, Informationsteknik | Avancerad | Universitetsadjunkt | 0 | 100 | Examinator, kursansvarig |
| Joachim von Hacht | Univ adjunkt | Datateknik, Informationsteknik | Grund | Universitetsadjunkt | 0 | 100 | Examinator, kursansvarig |
| Per Zaring | Univ adjunkt | Datateknik | Grund | Universitetsadjunkt | | 100 | Examinator, kursansvarig |
| Ants Silberberg | Tekn dr | Elektroteknik | Grund | Universitetslektor | 21 | 79 | Examinator, kursansvarig |
| Arne Dahlberg | Univ lektor | Datateknik, Informationsteknik | Grund | Universitetslektor | 0 | 100 | Examinator, kursansvarig |
| Björn von Sydow | Univ lektor | Datateknik, Informationsteknik | Grund, Avancerad | Universitetslektor | 0 | 70 | Examinator, kursansvarig |
| Christer Carlsson | Univ lektor | Datateknik, Elektroteknik, Informationsteknik | Avancerad | Universitetslektor | 0 | 100 | Examinator, kursansvarig |
| Claes Wernemyr | Univ lektor | Datateknik | Grund | Universitetslektor | | 100 | Examinator, kursansvarig |
| Lars-Eric Arebrink | Univ lektor | Datateknik | Grund | Universitetslektor | | 100 | Examinator, kursansvarig |
| Lena Peterson | Univ lektor | Datateknik | Avancerad | Universitetslektor | | 95 | Utbildningsområdesledare, examinator, kursansvarig |
| Rogardt Heldal | Univ lektor | Datateknik, Informationsteknik | Grund, Avancerad | Universitetslektor | 60 | 40 | Examinator, kursansvarig |

| Namn | Titel | Huvudområden | Nivåer | Tjänstebenäm. | Forsk | GRU | Roll |
|------------------|-------------|-----------------------------------|---------------------|--------------------|-------|-----|---|
| Roger Johansson | Univ lektor | Datateknik, Informationsteknik | Grund | Universitetslektor | 40 | 55 | Examinator, kursansvarig |
| Stefan Wirsenius | Tekn dr | Samhällsbyggnadsteknik | Grund | Universitetslektor | 80 | 20 | Examinator, kursansvarig |
| Sven Järner | Univ lektor | Matematik | Grund | Universitetslektor | 0 | 75 | Examinator, kursansvarig |
| Thomas Wernstål | Univ lektor | Matematik | Grund | Universitetslektor | 0 | 97 | Examinator, kursansvarig |
| Tomas Olovsson | Univ lektor | Datateknik, Informationsteknik | Avancerad | Universitetslektor | 75 | 25 | Examinator, kursansvarig |
| Uno Holmer | Univ lektor | Datateknik, Informationsteknik | Grund | Universitetslektor | 0 | 100 | Examinator, kursansvarig |
| Wolfgang Ahrendt | Univ lektor | Datateknik, Informationsteknik | Grund, Avancerad | Universitetslektor | 50 | 50 | Programansvarig IT, examinator, kursansvarig |
| Ulla Dinger | Univ lektor | Matematik | Grund | Universitetslektor | | | Examinator, kursansvarig |