International Graduate School in Computer Science

The Computer Science Graduate School is aimed at both industry and the academic world and covers the following subject areas:
- Computer Science
- Computer Systems
- Information Science and Media
- Computational Linguistics
- Economic Information Systems
- Information Systems Development
- Engineering Information Systems

The research environment is strongly influenced by cooperation at both departmental and international levels and the Department is regularly visited by guest professors and graduate students from international study programs.

The aims of the graduate school are the following:
- The primary aim of educating competent researchers at this high-quality establishment. Methods of continually assessing progress and results and proposing improvements to achieve this end are considered essential.
- The graduate school emphasizes the value of an integrated course of education in an area of importance for Swedish industry. The aim is to provide the student with broad competence: on completion of studies the student will have deep insights into his or her area of study as well as being well-oriented in the state of the art in related fields.
- The department has 14 laboratories and all graduate students belong to one of them. This provides an environment where the student, supported by advisors, formulates and produces his or her thesis as part of the requirements.
- In addition to a main advisor each graduate student has two deputy advisors. The advisory group can provide the student with a wider range of support than is possible with just one advisor.
- The course-work pursued is of central importance in gaining broad competence. The Department offers a well-established program of about 30 courses per year. These are often of an interdisciplinary character, thus the range is not limited to the student's particular lab, but is of relevance to the Department as a whole. In addition to courses of a more "technical" nature, others are given in research methodology and scientific writing. Each laboratory also runs courses specific to its range of interests.
- As a consequence the study program promotes communication between students pursuing different interests. Seminar series, graduate student conferences, information and assessment meetings also stimulate collaboration.
- In addition to traditional graduate studies the Department's aims have for many years included the further education of teachers and lecturers at regional University Colleges, as well as continuing education for applicants from industry.

Further information concerning the Graduate School of Computer and Information Science can be obtained from

Lillemor Wallgren
Administrator of Graduate Studies
Department of Computer and Information Science
Linköpings universitet, S-581 83 Linköping Sweden
Phone: +46 13281480 (281000) • Telefax: +46 13142231 • Internet: lew@IDA.LIU.SE
Department of Computer and Information Science
<table>
<thead>
<tr>
<th>Course</th>
<th>Course Literature</th>
<th>Author</th>
<th>Jan</th>
<th>Febr</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
<th>Day</th>
<th>Time</th>
<th>Lokaler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity Theory</td>
<td>1. Introduction to the theory of Complexity</td>
<td>1. D.P. Bouvet, P. Crescenzi</td>
<td>This is an undergraduate course - separate schedule</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Algebra and Object Oriented Mathematical Modeling</td>
<td>1. TCP/IP Illustrated, Volume 1: The Protocols</td>
<td>1. W. Richard Stevens</td>
<td>Start on March 5</td>
<td>Tues</td>
<td>9-12</td>
<td>Estraden</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed Objects</td>
<td>1. Software Engineering</td>
<td>1. Ian Sommerville</td>
<td>Start on Jan 5</td>
<td>Mon + Wed June 5, 12</td>
<td>9-12</td>
<td>20/5 Belöningen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Retrieval and Information Filtering</td>
<td>1. Introduction to Modern Information Retrieval</td>
<td>1. Salton, Gerard; McGill, Michael J.</td>
<td>Start on March 6</td>
<td>Wed</td>
<td>10-12</td>
<td>Eliten</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction to Description Logics</td>
<td>1. Investing in Information Technology: Managing the Decision-making Process</td>
<td>1. Geoff Hogbin, David V. Thomas</td>
<td>Start on Jan 18</td>
<td>Mon</td>
<td>13-16</td>
<td>Elogen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Graduate Courses Spring 1996

<table>
<thead>
<tr>
<th>Course</th>
<th>Course literature</th>
<th>Author</th>
<th>Jan</th>
<th>Febr</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
<th>Day</th>
<th>Time</th>
<th>Lokaler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methodology of Research in Computer Science</td>
<td>Credit: 3p (+3p)</td>
<td>Jan Febr</td>
<td>Jan 22</td>
<td>Febr 5, 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mon</td>
<td>13-15</td>
<td></td>
</tr>
<tr>
<td>Modeling and Control of Logical Discrete Event Systems</td>
<td>Credit: 2p - 4p</td>
<td>Staffan Bonnier</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13-15 Elien</td>
</tr>
<tr>
<td>Modelling och värdering av systemutvecklingsmetoder</td>
<td>Credit: 5p</td>
<td>Göran Goldkål, Anders G., Nilsson</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Database Design</td>
<td>Credit: 4+2p</td>
<td>Tore Risch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-Time Systems</td>
<td>Credit: 3p (+2)</td>
<td>Anders Törne</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utredningsmetodik och Kvantitativa metoder</td>
<td>Credit: 10p</td>
<td>Birger Rapp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Course Details

1. **Methodology of Research in Computer Science**
   - **Author:** Jan Febr
   - **Credit:** 3p (+3p)

2. **Modelling and Control of Logical Discrete Event Systems**
   - **Author:** Staffan Bonnier
   - **Credit:** 2p - 4p

3. **Modelling och värdering av systemutvecklingsmetoder**
   - **Credit:** 5p
   - **Authors:** Göran Goldkål, Anders G., Nilsson

4. **Network Database Design**
   - **Credit:** 4+2p
   - **Author:** Tore Risch

5. **Real-Time Systems**
   - **Credit:** 3p (+2)
   - **Author:** Anders Törne

6. **Utredningsmetodik och Kvantitativa metoder**
   - **Credit:** 10p
   - **Author:** Birger Rapp

### Course Schedules

- **Complexity Theory**
  - **Schedule Details:**
    - **Thursday:** Mar 21 8-10 FO5
    - **Friday:** Mar 22 10-12 FO7
    - **Monday:** Mar 25 13-15 FO7
    - **Wednesday:** Mar 27 15-17 FO4
    - **Friday:** Mar 29 10-12 FO6
    - **Tuesday:** Apr 2 8-10 FO7
Contents

Graduate School of Computer and Information Science Graduate Program 1995-1996.............1

General Graduate Courses Fall 1995:

Advanced Computer-based Learning Environments, 3p+1p, B Wasson.......................3
Activity Theory: Applications in Computer Science, 5p, T Timpka ..........................4
Applied Rewriting, 3p, J Maluszynski, U Nilsson .................................................5
Aspects of Scientific Writing, 3p, U Nilsson ..............................................................6
Avhandlingsanalys-Informationssystem, 2-3p, A.G Nilsson .......................................7
Cognitive Semantics, 4+1p, R Hirsch ...........................................................................8
Configuration Management, 3p, P Fritzson, B Magnusson ........................................9
Formal specification of reactive systems: theory and practice, 3p,
S Nadjm-Tehrani, J Malec ............10
Information Retrieval 1: Traditional and newer approaches, 3p, R Hjerppe ..........11
Introduction to Research Methodology in Computer Science, 2p, S Hägglund ....12
Kunskapsutveckling-teori, projektering, genomförande, 3p (del 1) + 3p (del 2)
Göran Goldkuhl ..................................................13
Kvalitativ analys och teoriutveckling, 3p, G Goldkuhl ............................................15
Logical- and Performance Debugging of Parallel Programs, 4p, O Hansen,
P Fritzson .....................................................................................................................16
Management of Information Technology, 2,5+2,5p, T Falk ..................................19
Principles of Modern Database Systems, 4+2p, T Risch ........................................20
Topics in Software Engineering, 3p, B Lennartsson ...............................................21

General Graduate Courses Spring 1996:

Complexity Theory, 4p, P-O Fjällström .................................................................23
Computer Algebra and Object Oriented Math. Modeling, 3+1p, P Fritzson ........24
Computer Network Services and Protocols, 5p, P Fritzson ..................................25
Distributed Objects, 3+1p, J Fagerström ...............................................................27
Ekonomisk styrning, 5p, N-G Olve .................................................................28
Evaluative Methodology, 3+2p, B Lennartsson ......................................................29
Information Retrieval and Information Filtering, 3p, N Shahmehri .......................31
Introduction to Description Logics, 4+1p, N Shahmehri ........................................32
IT-ekonomi och informationsekonomi, 3p, B Rapp, T Falk, N-G Olve .................34
Mathematical System Theory, 3p, J Malec .............................................................35
Methodology of Research in Computer Science, 3p (+3p) S Hägglund ...............37
Modeling and Control of Logical Discrete Event Systems, 2p-4p, S Bonnier,
J Maluszynski ..........................................................38
Modellering och värdering av systemutvecklingsmetoder, 5p, G Goldkuhl,
A.G Nilsson ............................................................39
Network Database Design, 4+2p, T Risch ...............................................................40
Real-Time Systems, 3p (+2), A Törne .................................................................42
Utredningsmetodik och Kvantitativa metoder, 10p, B Rapp ..................................43
General Information about Graduate Studies in Computer and Information Science

Graduate studies at the department consists of courses and project participation. The course programme is organized at the department level as regular courses, each of which is given approximately every second or third year (if possible), and occasional courses which depend on the profile and interests of current faculty and visiting scientists. The programme covers the areas: Computer science, Computer Systems, Library and Information Science, Economic Information Systems, Computational Linguistics, Engineering Information Systems and Information Systems Development.

About 100 students participate in the graduate programme, and may choose among about 30 courses given each year. The courses and seminars are normally given in English (unless all participants are fluent in Swedish). About 10 of the students and about 25% of the teachers in the graduate programme have foreign citizenship or origin, which makes the programme activities very international and English the language of the programme.

The programme leads to one of the following degrees:

**Licentiate of technology or philosophy.** The requirements include 40 points (one point equivalent to one week full time studies) of completed courses and 40 points thesis work. For a licentiate of technology, a master of engineering ('civilingenjör', 4.5 years of study) is normally assumed as a prerequisite.

**Doctor of technology or philosophy.** The requirements are 80 points courses and 80 points of thesis work. Most of the Ph.D. students take the licentiate degree as an integral part of their doctoral studies.

The Research Committee, headed by Sture Hägglund, is responsible for the organization and implementation of the graduate programme. The members of the research committee are mainly senior researchers, but there are also representatives for the graduate students, and for the technical and administrative staff. As an executive, there is one director of graduate studies (forskarstudierektor). However, most of the administration and organization rests upon the secretary of research (Lillemor Wallgren). Most graduate students are employed by the department, full time. Their responsibilities comprise of, for example, assisting in undergraduate courses and other internal assignments of the laboratories, up to about 15 - 30% of their time. The rest of the time is spent on courses and thesis project.
Graduate Programme
in Computer and Information Science 1995/1996

This program contains the following types of courses:

- Graduate Courses at the Graduate School of Computer and Information Science
- Graduate Courses in the School of Engineering
- Recommended Master Courses
- Laboratory-Oriented Courses and Activities

It also includes presentations of
- Research Organization
- Faculty

In addition to the above mentioned courses the following activities take place in the Graduate School of Computer Science and are strongly recommended for all graduate students:

Introduction meeting for graduate students.

An Introduction meeting will take place on Tuesday, August 29, 1995 at 13.15 in Estraden, E-building 1st floor.

Main seminar series on Tuesdays at 13.15.

The seminars are announced by e-mail, in the IDA-Kuriren, and occasionally by special announcement. They are usually given in Estraden, E-building, 1st floor or Belöningen, B-building, 1st floor.

Departmental coffee-breaks on Tuesdays (IDA-fika)

Current information, short presentations of new arrivals and visitors in the department, descriptions of trips and conferences etc. are given every Tuesday at 12.30 in the coffee area, E-building, 1st floor.

Further information concerning the contents of this program can be obtained from Lillemor Wallgren, tel- 013 28 14 80 and Per-Olof Fjällström, tel. 013-28 24 12, or for a particular course from the person responsible for the course.

Linköping, June 14, 1995

Lillemor Wallgren
Department of Computer and Information Science
Linköping University
S-581 83 Linköping.
Phone: 013-281480, Fax: 013-142231, E-mail: lew@ida.liu.se
Advanced Computer-based Learning Environments

Recommended for: Graduate students in all areas

Lectures: 24 h

The course last ran: New course.

Goal: To introduce and investigate advanced computer-based learning environments as a research area.

Prerequisites: No specific prerequisites.

Organization: A combination of lectures, seminars, and group work

Contents: Topics such as:
Knowledge-based Education and Training Systems
Intelligent Tutoring Systems; Intelligent Learning Environments; Adaptive Learning Environments; Automated Instructional Design, Intelligent On-line Assistance, ...
Simulation-based Learning Environments
Educational Multimedia
Pedagogical and Psychological Issues (e.g. situated learning, constructivism, motivation...)
Computer Supported Cooperative Learning
Computers as Cognitive Tools
TeleTeaching (e.g. Virtual Classrooms / Campuses; Distance Learning; Internet-based Learning)

Literature: Articles and book chapters.

Teachers: Barbara Wasson
guest lecturers

Examiner: Barbara Wasson

Schedule: September - December 1995

Examination: Active seminar participation.
Term paper or programming assignment.

Credit: 3 points + 1
Activity Theory: Applications in Computer and Information Science

Recommended for: Lectures: 20 h
Graduate students in applied Computer and Information Science.

The course last ran:
New course

Goals:
To introduce Activity Theory as a framework for study of Information Systems.

Prerequisites:
Registrated as a doctoral student.

Organization:
Lectures, literature studies, group reports, term paper.

Contents:
Basic concepts of Activity Theory; Theoretical foundation; History of Activity Theory; Application in other areas; Applications in Computer Science; Case examples.

Literature:
To be announced.

Teachers:
Yrjö Engeström
Toomas Timpka

Examiner:
Toomas Timpka

Schedule:
October 1995

Examination:
Term paper

Credit:
5 points
Applied Rewriting

Recommended for: Graduate students in computer science and systems

Lectures: 28h

The course last ran: 1993/94

Goals: The objective of the course is to give a uniform view of different rewriting formalisms applied in different context. Such a general theory of rewriting provides a platform for studying and classifying various paradigms of computation.

Prerequisites: Formal languages

Organization: Lectures

Contents: A general concept of rewriting. Computing functions by rewriting (lambda-calculus and combinatory logic). Abstract data types. Computing relations by rewriting. We show that the well-known concept of context-free rewriting (grammar) extends naturally to logic programs, attribute grammars, and two-level grammars. In this way we get a unified perspective of grammars (capable to describe type 0, type 1 and type 2 languages) and programs. Defining semantics of programming languages in terms of rewriting systems. Foundations of integration of functions and relations. Equational unification and various kinds of narrowing in the perspective of integration of logic programming and functional programming.

Literature: Collection of papers and book chapters.

Teachers: J Maluszynski & U Nilsson

Examiner: J Maluszynski

Schedule: November 1995 - March 1996

Examination: Homework and active participation

Credit: 3 points
Aspects of Scientific Writing

Recommended for: Everyone

Lectures: ~15h

The course last ran: 1994/95

Goals:
To envisage problems and give practical hints on how to improve the writing.
To help speed up the writing process and prepare participants for potential problems.
To increase awareness of the writing process and initiate further discussions.

Prerequisites:
The course is given in Swedish

Organization:
Lectures (50%), group activities (50%)

Contents:
Scientific writing, the writing process, language and content, reviewing, the scientific paper, typography, the dissertation.

Literature:

Teachers:
Invited lecturers

Examiner:
Ulf Nilsson

Schedule:
Mid-September through October 1995

Examination:
Homework and active participation

Credit:
3 points

Note:
The course will precede the IDA Conference in Computer and Information Science, held in November 1995.
General Graduate Courses Fall 1995

Avhandlingsanalys - Informationssystem

Recommended for: Doktorander i informationssystemutveckling, ekonomiska informationssystem och informatik

Lectures: 16 h

Goals:
Öka förmågan att granska godkända lic- eller doktorsavhandlingar inom området informationssystem som lagts fram vid ett svenskt universitet/högskola.

Prerequisites:
Inga särskilda krav.

Organization:

Contents:

Literature:

Teachers:
Anders G. Nilsson

Examiner:
Anders G. Nilsson

Schedule:
November - December 1995

Examination:
Skriftlig rapport (uppsats) om egen avhandlingsanalys. Seminarier.

Credit:
2-3 p
Cognitive Semantics

Recommended for: Graduate students

The course last ran: This is new course

Goals: Critical introduction to cognitive semantics in relation to fundamental questions in the semantics of natural languages- i.e. the relation between language and cognition, language and communication.

Prerequisites: Equivalent of Linguistics Basic Course (C-program)

Organization: Lectures and discussion seminars with active participation from course members.

Contents: Linguistic categories, Prototype theory, Stereotypes, Concepts, Mental images, Mental models, Mental representation, Cognitive discourse semantics.

John Taylor (1989) *Linguistic Categorization Prototypes in Linguistic Theory*  
Peter Gärdenfors (1992) *Blotta Tanken*  
Journal Articles

Teachers: Richard Hirsch

Examiner: Richard Hirsch

Schedule: September 1995 - Spring 1996

Examination: Written or oral exam and course paper

Credit: 4 + 1 points
Configuration Management

Recommended for: Lectures: 20h
Graduate students in Computer Science

The course last ran:
New course

Goals:
To learn common terminology and organization techniques, experience from using traditional and modern tools. Acquaintance with current research in this area.

Prerequisites:
Programming courses, software engineering theory or practice, experience from SW projects.

Organization:
Lectures + self study + projects + presentations.

Contents:
Introduction and background:
- Traditional terminology, common systems and tools.
- Why configuration management (CM) is a problem.
- Work organization in large organizations, large projects.
Modern CM systems
- Functionality and use.
Current research in CM
- CM for highly structured data such as in CAD/CAM systems
- CM for teams of programmers.

Literature:
Collection of articles.

Teachers:
Boris Magnusson

Examiner:
Peter Fritzson, Boris Magnusson

Schedule:
October - November 1995

Examination:
Presentations + homework

Credit:
3 points
Formal specification of reactive systems: theory and practice

Recommended for: Lectures: 24 h
Computer Science and Systems students

The course last ran: New course

Goals:
To present the theory of reactive system specification based on communicating state machines (e.g. various sorts of transition systems or statecharts) and to give opportunity to gather a hands-on experience of advantages and drawbacks of using a commercial tool (Statement) developed for this purpose.

Prerequisites:
Formal Languages and Automata Theory (TDDA 89)

Organization:
A series of 12 meetings: lectures, seminars and an introductory lab.

Contents:
Theory behind the language of Statecharts;
Semantics for variants of Statecharts, Connections to transition systems;
Formal analysis and approaches to verification
Selected commercial tool: Statemate

Literature:
Articles distributed before and during the course.

Teachers:
Simin Nadjm-Tehrani, Jacek Malec

Examiner:
Simin Nadjm-Tehrani, Jacek Malec

Schedule:
October - December 1995

Examination:
Presentation of seminar papers and practical assignments with documentation.

Credit:
3 points
Information Retrieval I: Traditional and newer approaches

Recommended for: Graduate Students

Lectures: 24 h

The course last ran: 1990/91

Goals: Provide an overview of traditional issues and solutions in information retrieval.

Prerequisites: None

Organization: Reading assignment a will be specified or distributed one week before each meeting. Students will take turns in presenting the contents of the reading assignment and in leading the discussions of issues arising from it.

Contents: In Information Retrieval (IR) some of the main concerns are:

- the design of databases from a descriptive and a technical point of view
- the design and use of vocabularies for description as well as query formulation
- the translation of “documents” and information requests to matchable items
- methods of matching, and file organizations
- interfaces and interaction in searching
- methods for evaluation

Literature: To be decided later

Teachers: Roland Hjerppe

Examiner: Roland Hjerppe

Schedule: October - December 1995

Examination: A written report on a chosen topic

Credit: 3 points
Introduction to Research Methodology in Computer Science

Recommended for: New graduate students.

The course last ran: Fall 1994.

Goals: To give an introduction to the philosophy of science, the special characteristics of computer science research and to discuss practical aspects of graduate studies and scientific activities.

Prerequisites: None.

Organization: Lectures and seminars.


Literature: Chalmers: *What is this thing called science?* Sindermann: *Survival strategies for young scientists.* Lecture Notes.

Teachers: Sture Hägglund, et al.

Examiner: Sture Hägglund.

Schedule: September - November 1995

Examination: Written examination and seminar activity.

Credit: 2 points
Kunskapsutveckling - teori, projektering, genomförande

Recommended for: Forskarstuderande inom informationssystemutveckling, samt andra med intresse för kunskapsteori och forskningsmetodik.

Lectures: 24 h

The course last ran: 1994/95

Goals: Ge översikt över kunskapsteoretiska och forskningsmetodologiska grunder för empiriska studier, samt öka förmågan att planera och genomföra sådana studier.

Prerequisites: Inga särskilda krav.


Contents: Kursen fokuserar forskningsarbete som kunskapsutveckling och särskilt problemställningar inom samhällsvetenskaplig kunskapsbildning.


Genomförande av empiriska undersökningar: Olika typer av undersökningar (klassificering, strategier, tillförlitlighet). Olika undersökningsmetoder (främst intervju, observation, källanalys).

Literature:
Föllesdal, Wallöe, Elster: Argumentationsteori, språk och vetenskapsfilosofi, Thales
Lundahl, Skärvad: Utdningsmetodik för samhällsvetare och ekonomer, Studentlitteratur
Repstad: Nä rhet och distans, Studentlitteratur
Goldkuhl: Kunskapande, kompendium, IDA
Ytterligare artiklar i samhällsvetenskaplig metodik

Teachers: Göran Goldkuhl

Examiner: Göran Goldkuhl

Schedule: September - November 1995
Examination:
Utförande av arbetsuppgift som dokumenteras i skriftlig rapport. Arbetsuppgiften bör i första hand bestå av utförande av en kunskapsprojektering med egen vald inriktning (del 1). Arbetsuppgiften kan fortsättas med genomförande av en mindre empirisk undersökning. Resultat från denna undersökning ska dokumenteras tillsammans med särskild analys av forskningsmetodik och kunskapskaraktärisering (del 2). Deltagande på seminarier.

Credit:
3 p (del 1) + ca 3 p (del 2)
Kvalitativ analys och teoriutveckling

Recommended for: Lectures: 24 h
Doktorander med intresse för kvalitativa undersöknings- och analysmetoder

The course last ran: Ny kurs.

Goals:
Öka kunskap om kvalitativ analys av empiriska data och hur empirgrundande kategorier och teorier kan genereras.

Prerequisites:
Grundläggande kurs i forskningsmetodik. Kursen förutsätter grundläggande kunskaper i kvalitativ undersökningsmetodik.

Organization:
Koncentrera föreläsningar. Eget arbete med tillämpningsuppgifter. Seminarier.

Contents:

Literature:
AL Strauss: *Qualitative analysis for social scientists*, Cambridge University Press

Teachers:
Göran Goldkuhl

Examiner:
Göran Goldkuhl

Schedule:
September - December 1995.

Examination:
Skriftliga rapporter. Seminarium.

Credit:
3 p
Logical- and Performance Debugging of Parallel Programs

Recommended for: Graduate students in Computer Science

Lectures: 30h

The course last ran: New course. The previous course on debugging 1993 was on automated debugging.

Goals:
The participants should gain a working knowledge of concepts of debugging and performance analysis with special emphasis on software for parallel and distributed architectures.

Prerequisites:
Knowledge and experience of programming:

Organization:
Lectures + presentations + exercises

Contents:
Introduction to program debugging
- Logical Debugging
  - Purpose
  - Requirements
  - Problems
- Performance Debugging
  - Purpose
  - Requirements
  - Problems
- Introduction to realization techniques to be considered
  - Modes of operation
    (On line, off line...)
- Interacting with the program execution
  -(Program instrumentation, data collection, execution control)
  - User interaction
  -(Data transformation, filtering, data presentation,
  - graphical user interface)
Program interaction (instrumentation)
- Execution related event triggering
- Execution states to be considered
- Global states of execution in parallel systems
- Source code instrumentation
- Object code instrumentation
- Hardware support
- Data related event triggering
- Data to be considered
- Source code instrumentation
- Object code instrumentation
- Hardware support
- Producing statistical information
- Data to be produced
- Instrumentation techniques
- Hardware support
- Producing hardware related information
  - Data to be produced
  - Requested hardware support
- Execution control
  - Influencing the state of execution
  - Changing program data
- Standardization issues
  - PTRACE with extensions

Data collection and storage
- Controlling data collection and storage
  - On line
  - Off line
  - Event triggered
- Recording of event traces
  - Local trace collection
  - Hardware supported trace collection
  - Trace recording on distributed file systems
- Producing and recording of derived information
  - Accessing statistical information
- Local data reduction
- Global data reduction
- Standardization issues
  - SDDF
  - TDL

Data transformation
- Translation between high level languages and machine code
  - Source code lines
  - Data structures
  - Functions and Methods
  - Threads and Processes
  - Program modules
- Data reduction
  - Filters
  - Data base approaches
- Standardization issues
  - Symbol table formats
  - COFF
  - DWARF

User interface
- Levels of user interfaces
- Tool construction sets
- Interfaces with strong user guidance
- Design principles for debugging interfaces
- General principles (e.g. Source code orientation)
- Support for data parallelism
- Support for functional parallelism
- Presentation of event related data
- Space-Time diagrams
- Focusing on relevant information
- Presentation of statistical data
- Selection of relevant information
- Data presentation
- Massive parallelism
- Problems
- Solutions
- Browsing and zooming
- Automated focusing

Examples
- DETOP
- PATOP

Literature
Papers & Manuals

Teachers:
Olav Hansen

Examiner:
Olav Hansen, Peter Fritzson

Schedule:
September - October 1995

Examination:
Written exercises and programming exercises.

Credit:
4 points
Management of Information Technology

The course last ran:
New course.

Organization:
The course is built on lectures, seminars and case studies. There will be individual as well as group assignments on readings. During the course you are expected to prepare for case studies and submit individual written comments.

Contents:
The theme of the course is the role of modern information technology in business and society and its strategic and organisational consequences.

Literature:


Teachers:
Thomas Falk

Examiner:
Thomas Falk

Schedule:
October and December 1995

Examination:
There will be a final written exam, accounting for 80% of your grade. Your preparation of the case studies will account for the remaining 20% of the grade.

Credit:
2,5 + 2,5 points.
Principles of Modern Database Systems

Recommended for: Lectures 32h
Graduate students

Goals:
To present the fundamental concepts, theories and realizations of modern database technology and systems, with a concentration on modern Object-Oriented (OO) database systems, Query Processing, and Active Databases.

Periodicity:
Once per Year.

Prerequisites:
Undergraduate courses in computer science. Basic database course preferred, but not required.

Organization:
Lectures and seminars covering the core material of the course. Extra points for programming project or seminar papers.

Contents:
- Overview: Traditional Gibbons
- DBMSS architectures
- Data Models (e.g. relational, OO, functional)
- Object Data Management Concepts
- Query processing (OO and relational)
- Concurrency and Recovery
- Versioning
- Database Performance and Benchmarks
- Deductive Databases
- Active Databases
- Temporal Databases
- Main-memory Databases

Literature:
R.G.G. Catell: *Object Data Management* + handouts

Teachers:
Tore Risch

Examiner:
Tore Risch

Schedule:
November 1995 - January 1996

Examination:
Written exam, Seminar Papers, and Small Programming Project

Credit:
4+2 points
Topics in Software Engineering

Recommended for: Graduate students at the department

Lectures: 20h

The course last ran: New contents

Goals: To give an overview of concepts and models in Software Engineering, and to present selected relevant activities in the area.

Prerequisites: None.

Organization: Individual readings, lectures, guest seminars, and paper presentations.

Contents: Overview of software engineering. Software architecture and software reuse. Industrial development projects.


Teachers: Invited speakers, Peter Fritzson, Bengt Lennartsson.

Examiner: Bengt Lennartsson

Schedule: During 1995/96.

Examination: Term papers(s).

Credit: 3 points.
Complexity Theory

Recommended for: Computer Science students.

Lectures: 30 h

The course last ran: A course with similar contents was given in spring -93

Goals: The systematic study of computational complexity theory has developed into one of the central and most active research areas of computer science. The aim of this course is to present the most significant results obtained in this area of research.

Prerequisites: Basic course on design and analysis of algorithms.

Organization: Two lectures weekly, obligatory homework.

Contents:
1. Elements of computability theory
2. Complexity classes
3. The class P and NP
4. The complexity of optimization problems
5. Beyond NP
6. Space-complexity classes
7. Probabilistic algorithms and complexity classes
8. Interactive proof systems
9. Models of parallel computers and parallel algorithms


Teachers: Per-Olof Fjällström.

Examiner: Per-Olof Fjällström.

Schedule: Spring 1996.

Examination: Obligatory homework. Written solutions to homework must be submitted and each student must be prepared to orally present her solutions.

Credit: 4 points.
Computer Algebra and Object Oriented Mathematical Modeling

Recommended for: Graduate students in Computer Science

Lectures: 22h

The course last ran: New course. The last graduate computer algebra course ran at least 9 years ago.

Goals: The participants should gain a working knowledge of concepts in computer algebra systems, how to use these systems, and how to construct object oriented mathematical models.

Prerequisites: Knowing the basic concepts of object orientation, and having.

Organization: Lectures + presentations + exercises

Contents:
1. Computer Algebra Languages
   - Abstraction concepts
   - Function and rule invocation
   - Algebraic transformations
   - Systems: Mathematica, Maple, Axiom
2. Object Oriented Mathematical Modeling
   - Modeling styles
   - Interfaces, inheritance & encapsulation
   - Interactions between computer algebra and OO
   - Systems: Dymola, ObjectMath, etc.

Literature: Articles

Teachers: Peter Fritzson, invited speakers

Examiner: Peter Fritzson

Schedule: Spring 1996

Examination: Programming, transformation and modeling exercises using available systems.

Credit: 3 + 1 points
Computer Network Services and Protocols

Recommended for: Graduate students in Computer Science and Computer Systems

Lectures: 36 h

The course last ran: New course

Goals:
To give an understanding of principles for design and implementation of network services and the underlying protocols. The principles are exemplifies by in-depth studies of existing Internet protocols.

Prerequisites:
Basic knowledge about computer networks (e.g. from the undergraduate course "Computer networks") is preferable, but not necessary. A working knowledge of the C programming language is necessary for the exercises.

Organization:
Lectures and practical exercises.

Contents:
- Introduction to computer networks. Basic principles and terminology (the OSI model etc.)
- The physical and link layers. Examples: IEEE 802, FDDI, ATM, ISDN.
- Network security.
- The Internet: organization and standards.
- Protocols for the network and transport layers. Examples: IP, UDP, TCP, ICMP, ARP, RARP.
- Programming interfaces to such protocols. Examples: BSD sockets, TLI.
- Point-to-point protocols. Example: SLIP, PPP.
- Simple application protocols. Example: TELNET
- Routing and routing protocols. Examples: OSPF, RIP.
- The domain name service (DNS.)
- Network management tools and protocols. Example: SNMP. (guest lecturer)
- Protocols for E-mail. Examples: SMTP, POP3, IMAP.
- Standards for message encoding. Example: MIME.
- Protocols for other information services. Examples: FTP, NNTP, HTTP.
- Remote procedure calls. Principles, protocols, programming interfaces and tools. Examples: Sun RPC and XDR.
- Distributed file systems. Example: NFS
- Authorization. Example: Kerberos
- SUNET: technology, structure, organisation, and history. (Guest lecturer)
- Summary of the course.

Literature:
A textbook on computer networks (e.g. Comer: Internetworking with TCP/IP) and protocol specifications (RFCs.)

Teachers:
Lars Viklund, Niclas Andersson, and invited lecturers.
Examiner:  
Peter Fritzson

Schedule:  
Spring 1996

Examination:  
Three individual design and implementation exercises.

Credit:  
5 points
Distributed Objects

Recommended for: Graduate students in Computer Science

Lectures: 20h

Goals:
The participants shall understand the coming integration of object-oriented and distributed systems. In particular, the participants will understand the importance of standardized middleware used to isolate operating systems, networks and programming languages from object-oriented applications.

Prerequisites:
The participants must understand the object-oriented model and have a general understanding of distributed systems. A working knowledge of C or C++ is required for the extra graduate point project.

Organization:
Lectures + self study + project

Contents:
1. Distributed system architectures
   - variations on client-server
   - traditional ad OO distributed databases
   - distributed objects
2. Integration of systems using distributed objects
3. CORBA
   - The Common Object Model
   - IDL, Interface Definition Language
   - mappings to C, C++ and Smalltalk
   - Object brokering
4. Products
   - IBM SOM/DSOM
   - Digital’s ObjectBroker
   - IONA / Expersoft / SUN DOE and other systems
5. Component-based system development
   - OpenDoc
   - OLE 2
6. “The future distributed systems”

Literature:
Handouts of articles

Teachers:
Johan Fagerström

Examiner:
Johan Fagerström
Ekonomisk styrning

Recommended for: Forskare med baskunskaper i redovisning, kostnadsintäktanalys och organisation.

Lectures: ca 35 h

The course last ran: Hösten 1993.

Goals: Gemeinsamt utforska ekonomisk informationspotential som styrmedel internt i organisationer, med särskild betoning på målstyrning och decentralisering.

Prerequisites: Inga formella.

Organization: 5 heldagars undervisning under IDA/EIS:s forskarveckor.


Literature: Böcker av Johansson-Östman, Samuelson (red) och (prel.) Anthony-Dearden-Govindarajan; artiklar och enstaka kapitel ur andra skrifter, delvis i form av individuella referatuppdrag.

Teachers: Nils-Göran Olve.

Examiner: Nils-Göran Olve

Schedule: Våren 1996.

Examination: Presentationer och uppsatser under kursens lopp.

Credit: 5 poäng
Evaluative Methodology

Recommended for: Students interested in evaluation of new technology.

Lectures: 12h

The course last ran: New course

Goals: To give an introduction to qualitative methods for the evaluation of new (software) technology.

Organization: Lectures, seminars, paper presentations, exercises.

Contents: Planning the evaluative research. Methods for data collection; interviews, etc. Methods for data analysis. Optional exercise.

Literature: To be decided.

Teachers: Invited speakers, Toomas Timpka, Göran Goldkuhl, Bengt Lennartsson.

Examiner: Bengt Lennartsson

Schedule: Spring 1996

Examination: Term papers(s) for 3 points. A small exercise (planning, collection, analysis, conclusions, report). Additional 2 points.

Credit: 3 + 2 points

Schedule: Spring 1996
Examination:
Written exam and a programming exercise for extra graduate point(s).

Credit:
3 + 1 points
Information Retrieval and Information Filtering

Recommended for: Graduate Students. (Orienteringskurs C4)

The course last ran: New course

Goals:
The course goal is to give an introduction to information filtering techniques and the underlay­
ing technology (statistics, collaboration, natural language understanding, learning). Hands on experience with a few existing systems will be included. The course will include an introduction to the WWW and its relevance and role in information retrieval.

After the course, students should have an understanding of existing information filtering tech­niques, their limitations and possibilities. They will also have some experience with a few sys­tems.

Prerequisites:
None.

Organization:
A combination of lectures + seminars (prepared by the participants) + practical exercises.

Contents:
- Information Retrieval (IR)
- Connection between IR and Information Filtering (IF)
- Fundamentals of IF (Requirements)
- Basic Approaches to IF (IR, collaboration, group reviews, rule based, agent oriented, user modeling, natural language)
- Email and news filtering systems
- Machine learning of user preferences
- Agents: Standalone and Cooperating
- Privacy issues
- WWW including agents for searching the net and WWW as interface to DB

Literature:
Articles

Teachers:
IISLAB members

Examiner:
Nahid Shahmehri

Schedule:
Spring 1996

Examination:
Laboratory Assignments and a Term paper

Credit:
3 points
Introduction to Description Logics

Recommended for: Graduate Students. (Fördjupningskurs C4)

Lectures: 30 h

The course last ran: New course

Goals:
Description logics (or terminological logics or concept languages) are computationally attractive limited subsets of first-order logic. They are used for representing concepts and individuals in a given "world" and then drawing logical inferences from the given information. The inferences center around, but are not limited to, classification. There are a number of implemented description logic systems which are used for building a variety of applications including software management systems, planning systems, telephone configuration systems, and natural language understanding. Description logic systems are particularly suitable for applications where classification is a useful inference. At the conclusion of this course you will have:

- good understanding of the principles of description logics,
- used a number of different description logic systems, and compared their differences,
- understanding of the central algorithms used in description logic systems, in particular subsumption and classification,
- looked briefly at some areas of current development in description logics.

Prerequisites:
a basic course in logic (e.g. Logic for ida-ites, Logik grundkurs (TD2A15))

Organization:
Seminars (prepared by the participants) + labs

Contents:
- Introduction to the field, what description logics are, KL-ONE
- Representing knowledge in description logics
- The algorithms: classification, subsumption, other inferencing
- Completeness, Correctness, Complexity
- Revision and update of knowledge bases
- Systems: KRIS, LOOM, CLASSIC
- Some applications using description logics
- Connection to databases
- Hot topics: a.o. integrating different kinds of reasoning with description logics (e.g. defaults, part-of, time, actions)

Literature:
Articles

Teachers:
Patrik Lambrix

Examiner:
Nahid Shahmehri
General Graduate Courses Spring 1996

Schedule:
Spring 1996

Examination:
Seminar, lab assignments, summaries
Extra credit for larger project or term paper

Credit:
4+1 points
IT-ekonomi och informationsekonomi

Recommended for:
Ekonomiska informationssystem.

The course last ran:
Första gången.

Goals:
Att förstå sambandet mellan IT och ekonomiska effekter i företag och samhälle

Prerequisites:
Ekonomisk grundexamen.

Organization:
En betydande del av kursen kommer att bestå av tillämpningsövningar där deltagarna får arbeta med exempel på IT-ekonomi och informationsekonomi. Därtill kommer föreläsningar kring metodproblem och genomgångar.

Teachers:
Nils-Göran Olve
Thomas Falk
Birger Rapp

Examiner:
Nils-Göran Olve
Thomas Falk
Birger Rapp

Schedule:
Start våren 1996.

Examination:
För godkänt på kursen krävs ett aktivt deltagande i övningar samt genomförda övningsuppgifter.

Credit:
3 poäng
Mathematical Systems Theory

Recommended for: Computer Science and Systems students

Lectures: 24h

The course last run: 1992/93

Goals:
To give an introductory presentation of formal systems theory. To present systems theory as a metalanguage for engineering sciences. To present a generic formal definition and formulation of properties of dynamic systems. To illustrate connections between different application areas of systems approach.

The course in not intended to substitute any control theory or automata theory courses. It is rather intended to serve as a follow-up course, both for computer science students (C-line background) interested in basic control theory concepts, and computer systems students (D-line background) interested in basic theoretical computer science concepts.

Prerequisites:
In principle, no preliminary knowledge except elementary set theory and elementary algebra is required. However, knowledge of the material presented during one or more of the following courses might be helpful.

- Reglerteknik (TSRT 12,13,15,16,18)
- Control Theory (TSRT 35,38)
- Modelling (TSRT 20,50,64)
- Formal Languages and Automata Theory (TDDA 89)

Organization:
A series of 12 lectures, possibly with some classes.

Contents:

1 Definitions and classification
   -general system (any relation defined on $x_i \in V_i$);
   -input-output system (any relation defined on $(x_i \in A_i) \times (x_j \in B_j)$);
   -time system (a relation defined on $A^T \times B^T$);
   -dynamic system (a system which can be described using two families of functions (r-response family and state-transition family) fulfilling appropriate criteria);
   -classification based on assuming additional structure imposed on inputs, outputs, states, or time.

2 Problems studied within the systems theory framework
   -realization and minimal realization problems;
   -analysis of systems:
     -observability,
     -controllability,
     -stability;
   -composition and decomposition of systems;
   -goal-orientedness.
Literature:
Alternatively a compendium prepared by myself;
Literature recommended:
L. Padulo and M. A. Arbib *System Theory: a unified state space approach to continuous and discrete systems*, W.B.Saunders, Philadelphia, 1974

Teachers:
Jacek Malec

Examiner:
Jacek Malec

Schedule:
Spring 1996

Examination:
Written assignments (exercises) throughout the course. Term paper.

Credit:
3 points
Methodology of Research in Computer Science

Recommended for: Lectures: 16 h + Seminars 10-15 h
For graduate students

The course last ran: 1990

Goals: The goal of this course is to give the student an understanding for methodological problems in the process of performing research in general and in computer science in particular. The course also includes practical issues in connection with selecting a research problem, planning a project, carrying out the research and reporting the results.

Prerequisites: None

Organization: The course will have the character of an advanced seminar with presentations on selected topics, mainly by invited lecturers. Some letters will be used for a review of more practical issues in planning and carrying out research. There will also be a number of more discussion-oriented seminars, where specific papers or issues are treated. An additional 3 points may be obtained for writing an essay on a topic suggested or approved by the course leader.

Contents:
• Introduction to the theory of science
• Scientific paradigms
• Project conception and planning
• Scientific writing and publication
• Seminars with discussion of papers and dissertations from a methodological point of view

Literature: To be decided later. Articles and reports.

Teachers: Sture Hägglund, Roland Hjerpe, Erik Sandewall, invited lecturers

Examiner: Sture Hägglund

Schedule: Spring 1996

Examination: Active participation in lectures and seminars. A written summary of course material and other completed assignments.

Credit: 3 points (+3 points)
Modeling and Control of Logical Discrete Event Systems

Recommended for: Computer Science and systems graduate students.

Lectures: 20-30 h

The course last ran: New course

Goals: The field of discrete event systems has emerged to provide a formal treatment of many of the man-made systems such as manufacturing systems, communication networks, automated traffic systems, database management systems, and computer systems that are event-driven. The goal of this course is to present a general theory of logical discrete event systems, where the focus is on the order of events rather than on their occurrence times.

Prerequisites: Some degree of 'mathematical maturity'.

Organization: Seminars given by participants + invited speakers.

Contents:
- Introduction to Formal Language and Lattice Theory,
- Control under Complete Observation,
- Control under Partial Observation,
- Control of Non-Terminating Behavior.


Teachers: Staffan Bonnier, Jan Maluszynski

Examiner: Staffan Bonnier

Schedule
Spring 1996

Examination: Seminars + homework

Credits: 2p - 4p
Modellering och värdering av systemutvecklingsmetoder

Recommended for: Lectures: 40 h
Doktorander i informationssystemutveckling. Av intresse också för doktorander i ekonomiska informationssystem, informatik, datalogi.

The course last ran:
Ny kurs.

Goals:
Öka förståelsen om systemutvecklingsmetoder, och liknande metoder för verksamhetsutveckling, genom modellering och utvärdering. Öka kunskapen om metametoder för modellering och utvärdering av metoder.

Prerequisites:
Inga särskilda krav. Gärna någon grundläggande kurs i systemutveckling.

Organization:

Contents:
Metamodellering av metoder för system/verksamhetsutveckling;
Metodanalys/SIMM och andra metametoder för modellering av metoder; olika metodkomponenter i en metodmetod; utvärdering av metoder; kriterier för metodvärdering; NIMSAD och andra metodansatser för metodvärdering

Literature:
Artiklar

Teachers:
Göran Goldkuhl (metodmodellering),
Anders G Nilsson (metodvärdering)

Examiner:
Göran Goldkuhl, Anders G Nilsson

Schedule:
Spring 1996

Examination:
Skriftliga rapporter. Seminarium.

Credit:
5 p
Network Database Design

Recommended for: Graduate students
Lectures: 26 h + 10 h exercises

Periodicity: Once every second

Goals: Gain a good understanding of database design especially concerning transaction processing and reliability.

Prerequisites: Undergraduate courses in computer science. Basic database course.

Organization: Lectures and seminars covering the core material of the course. Extra points for book project or design project.

Contents:
- Network Database Applications
- Network Database Requirements
- AML, A Modelling Language
- System Model
- Operations of the Database
- Infrastructure Model (HW Model)
- Conceptual Models of Operations
- Logical Node Models
- Service Model
- Abstract Machine Model
- A Reliable Update
- Implementation Model
- Implementation of Concurrency Control
- Implementation of Recovery
- Implementation of Access Structures
- Change Management (Schema upgrades, software change etc.)
- Research topics

Literature:
J. Gray, Transaction Processing, Concepts and Architectures, A Reuter Lecture Notes
S-O Hvasshovd, A Tuple Oriented Recovery Method for Continuously Available Distributed DBMS on a shared nothing Multi-computer
Research articles
Modern Database Systems and Distributed Databases preferred.

Teachers: Mikael Ronström

Examiner: Tore Risch
General Graduate Courses Spring 1996

Schedule:
Spring 1996

Examination:
Written exam. Design project.

Credit:
4 + 2 points
Real-Time Systems

Recommended for: Computer Science and Systems graduate students.

Lectures: 24 h

Periodicity: Every second year

Goals: The course will give a survey to different research issues, methods and state of the art in the design of real time systems.

Prerequisites: Processprogramming TDDA21 or corresponding prerequisites Basic knowledge in finite automata, logic and operating systems.

Contents:
- Fault tolerance, atomic actions and exception handling
- Scheduling and Resource Management
- Distributed real time systems
- Specification of real time systems - e.g., RTL, Petri Net Models
- Real Time Languages - synchronous and asynchronous
- RT-system design
- Case study

Literature
Probably from Burns&Wellings: *Real-Time Systems and their programming languages* together with articles and other excepts

Teachers: Anders Törne

Examiner: Anders Törne

Schedule: Spring 1996

Credit: 3p (+ 2p)
Utredningsmetodik och Kvantitativa metoder

Recommended for: Alla doktorander.

Lectures: 50-70 h

The course last ran: Våren och början av hösten 1995.

Goals:
- väl insatt i modern utredningsmetodik
- väl orienterad om kvantitativa metoder
- känna till centrala begrepp och teorier inom vetenskapsteori

Prerequisites:
Inga.

Organization:
Föreläsningar, seminarier, inlämningsuppgifter, tentamina.

Contents:
Vetenskapsteori, Kvalitativ undersökningsmetodik, Modelltänkande och systemanalys, Statistiska metoder, Simulering, Prognoser, Utvärdering och Presentationsteknik.

Literature:
Lekvall, Wahlbin, *Information för marknadsförare*
Savén, *Produktionssimulering*
Sellstedt, B., *Samhällsteorier*, Studentlitteratur 1992
Vedung, E., *Utvärdering i politik och förvaltning*, Studentlitteratur 1992
Kurskompendium

Teachers:
Ett flertal

Examiner:
Birger Rapp

Schedule:
Våren och början av hösten 1996.

Examination:
Aktivt deltagande på seminarierna, godkända inlämningsuppgifter och laborationer samt skriftliga och muntliga tentamina.

Credit:
10 poäng
Graduate Courses in the School of Engineering

**Presentationsteknik**
Fall 1995, Lars-Gunnar Petersson, IFM, tel 28 12 49

**Forskningsetik**
Fall 1995, Lars-Gunnar Petersson, IFM, tel 28 12 49

**Teknikhistoria**
Spring 1996, Lars Strömbäck, IKS, tel 28 18 13

**Vetenskapsteori**
Spring 1996, Ingemar Nordin, Tema H, tel 28 22 20
**Recommended Master Courses**

### C3-courses
- TDDA14 AI Programming
- TDDA21 Concurrent Programming
- TDDA37 Compiler Construction
- TDDA38 Database Technology
- TDDA41 Logic Programming
- TDDA43 Programming Theory
- TDDA99 Psychology of Communication
- TDDB40 Rewriting Systems
- TDDB42 Semantics of Programming Languages
- TDDB45 Computability and Complexity Theory
- TDDB60 Methodology of Program Development and Programming Development Project
- TGTU04 Leadership

### C4-courses
- TDDA12 System Development
- TDDA16 Representation of Knowledge in AI
- TDDA18 Natural-Language Processing
- TDDA30 Programming Theory II
- TDDA32 Design and Analysis of Algorithms
- TDDB33 Object-Oriented System Development
- TDDA66 Expert Systems-Methods and Tools
- TDDA67 Distributed Systems
- TDDB02 Software Quality
- TDDB10 Human-Computer Interaction
- TDDB15 Computer Aided Software Engineering for Development and Maintenance
- TDT541 Computer Networks
- TDT551 Advanced Computer Architecture

### SVL-courses
- CASE-verktyg i systemutveckling
- Datorteknik och datornät
- Informationssystem och MPS
- Människa-Datorinteraktion
- Objektorienterad systemutveckling
- Programmering i C och principer för programspråk
- Prototyping, systemutvecklingsmetoder och verktyg
- Samhällsvetenskaplig kunskapsbildning
Laboratory-oriented Courses and Activities

Like the graduate courses these are open for all graduate students at the department, but they are organized so as to have a direct link to activities in each laboratory. Additional course activities may be announced during the year.

The Laboratory for Complexity of Algorithms (ACTLAB)
Fixed time for lab meetings: Wednesdays 13-15.

The Laboratory for Application Systems (ASLAB)
The course Knowledge Acquisition, will be given by K Sandahl during spring 1996.
Fixed time for lab meetings: Thursdays 13-15.

The Laboratory for Computer Aided-Design of Digital Systems (CADLAB)
The course Digital ASIC Testing and Testable Design, will be given by Z Peng and K Kuchcinski during spring 1996 i.e. Tuesdays 10-12, weeks 6-15.
Fixed time for lab meetings: Thursdays 13-15.

The Laboratory for Engineering Databases and Systems (EDSLAB)
Fixed time for lab meetings: Thursdays 13-15.

Economic Information Systems (EIS)
No fixed time for lab meetings.

The Laboratory for Library and Information Science (LIBLAB)
Fixed time for lab meetings: Wednesdays 13-15.

The Laboratory for Intelligent Information Systems (IISLAB)
Fixed time for lab meetings: Wednesdays 13-15.

The Laboratory for Logic Programming (LOGPRO)
Fixed time for lab meetings: Wednesdays 15-17.

People, Computers and Work (MDA)
Fixed time for lab meetings: Wednesdays 13 - 15.

The Laboratory for Natural Language Processing (NLPLAB)
Fixed time for lab meetings: Wednesdays 15-17.

The Laboratory for Programming Environments (PELAB)
Fixed time for lab meetings: Thursdays 15-17.
Laboratory-oriented Courses and Activities

The Laboratory for Representation of Knowledge in Logic (RKLLAB)
Fixed time for lab meetings: **Wednesdays 13-15.**

The Real-Time Systems Laboratory (RTSLAB)
Fixed time for lab meetings: **Wednesdays 13-15.**

Information Systems and Work Contexts (VITS)
Fixed time for lab meetings: **Some Mondays 10-12.**
Research Organization and Laboratories

**IDA Board**
Anders Haraldsson, chairman
Erik Sandewall, vice chairman
Inger Emanuelson, executive secretary

**Undergraduate Teaching Committee**
Olle Willén
Britt-Marie Ahlenbäck

**Research Committee**
Sture Hägglund
Lillemor Wallgren

Areas for undergraduate courses

Research labs
Lab. leaders

The strategic and executive responsibilities for research projects and graduate education

A short presentation of the fourteen research laboratories at the department:

**ACTLAB – Laboratory for Complexity of Algorithms**
Per-Olof Fjällström

ACTLAB is concerned with the design and analysis of efficient algorithms (sequential and parallel) and data structures for combinatorial and geometric problems arising in computer science and the study of the inherent complexity of these problems in simple models of computation. One application area, studied in the context of CENIIT, is efficient algorithms for three-dimensional geometrical problems.

**ASLAB – Application Systems Laboratory**
Sture Hägglund

ASLAB is oriented towards the study of knowledge-intensive approaches to software development, including aspects of human-computer interaction. There are currently three subgroups dealing with cognition technology (Hägglund), knowledge engineering and software
Research Organization and Laboratories

engineering (Sandahl) and usability matters (Löwgren). Special areas of interest are industrial software quality and reliability, knowledge management, usability engineering and user interface design support, expert critiquing systems, knowledge acquisition, diagrammatic reasoning and intelligent tutoring systems.

CADLAB – Laboratory for Computer-Aided Design of Digital Systems
Krzysztof Kuchcinski

CADLAB concentrates its research activities on computer-aided synthesis and verification of digital systems, especially those involving very large-scale integrated circuits (VLST). The major concern is with the behavioural and structural aspects of digital systems specification, design, simulation, optimization, partitioning, synthesis and formal verification methods. Currently research projects in the areas of high-level synthesis, hardware/software codesign, and design for testability are being carried out in the laboratory.

EDSLAB – Laboratory for Engineering Databases and Systems
Tore Risch

EDSLAB does research on new database services and advanced applications in particular for supporting engineering information systems. The research is centered around the umbrella project AMOS - Active Mediators Object System, a next generation object-relational database mediator between applications and data sources. The laboratory also conducts database-oriented application projects for engineering support in cooperation with Swedish industry.

EIS – Economic Information Systems
Birger Rapp

EIS covers communication of information from people to/from systems or between people and the design of information systems supporting this communication. Research projects concern information support, agency theory, IT and organizational solutions, computer simulation for management training and decision support, business control and accounting and auditing.

IISLAB – Laboratory for Intelligent Information Systems
Nahid Shahmehri

IISLAB studies theory and methods for advanced information systems, including object-orientation, computer supported cooperative work, process modelling, workflow management, information retrieval and hypertext, and description logics. In a major implementation project a multi-user information system has been developed which supports parallel development of objects, historical information and automatic maintenance of the database via editing of structured objects.

LIBLAB – Laboratory for Library and Information Science
Roland Hjerppe

LIBLAB studies methods for access to documents and the information contained in the documents, concentrating on catalogues and bibliographic representations, and on the human factors of library use. Current interests are focused on i.a. document architecture issues, the
merging of information from libraries, archives and museums, spatio-temporal information and Geographic Information Systems, and formal approaches to the analysis of qualitative data.

LOGPRO – Laboratory for Logic Programming
Jan Maluszynski and Ulf Nilsson
LOGPRO long term research concentrates on the foundations of logic programming and relations to other programming paradigms and methodology. Presently focus is on query-optimization of deductive databases, proving dynamic properties of logic programs, and verification and synthesis of logic programs.

MDA – People, Computers and Work
Toomas Timpka
MDA conducts research into information system development and use in working life from the points of view of computer science, psychology, and social organization of work development. Within the MDA-group, activities at the Department of Computer and Information Science and the Medical Faculty have been coordinated to develop and evaluate experimental information systems.

NLPLAB – Natural Language Processing Laboratory
Lars Ahrenberg
NLPLAB is engaged in research on theoretical and applied natural language processing. The theoretical research is concerned with linguistic knowledge representation from the syntactic level to the discourse level and algorithms for parsing, generation and unification. The applied research covers natural language dialogue systems and tools for machine-aided translation and document generation.

PELAB – Programming Environments Laboratory
Peter Fritzson
PELAB conducts research in the area of tools and programming languages for software development and maintenance. Current projects include tools for semi-automatic bug location, debuggers for parallel languages, dependence analysis of programs, generation of efficient compilers from denotational semantic specifications, very high level languages and programming environments for scientific computing, and generation of parallel code for mathematical models.

RKLLAB – Laboratory for Representation of Knowledge in Logic
Erik Sandewall
RKLLAB conducts research on logic-based principles for the design of intelligent autonomous agents. This includes research on non-monotonic logics, logics for reasoning about action and change, fuzzy logic, algorithms for planning and temporal prediction, and related topics. It also includes research on methods for the systematic description of physical systems on a discrete level, and on architectures and tools for complex real-time systems.
RTSLAB - Real-Time Systems Laboratory  
Anders Törne  
RTSLAB research covers tools, methods and architectures for the design of software intensive real-time systems. In particular this includes timing analysis and design synthesis tools based on discrete modelling methods, database modelling of activities for control and simulation, and stratified real-time software architectures. The laboratory also conducts applicative research and case studies in cooperation with industry, for example in the automation and control area.

VITS - Development of Information Systems and Work Contexts  
Göran Goldkuhl  
VITS is a research group studying information systems development in relation to organisational aspects. Special research interest/projects on: Methods for change analysis, information requirements analysis and informations systems evaluation. Strategies for information systems architecture. Relations between methods and CASE tools (CASE shells).
Faculty Engaged in Graduate Study Program


Syntax, semantics and pragmatics of natural language; natural language understanding; natural language interfaces; machine-aided translation.


Formal methods in real-time system development, theory and practice of declarative programming.


Planning and temporal reasoning, algorithms and complexity for AI problems, representation and reasoning about knowledge.


Natural language processing, especially empirically based computational models of discourse. Cognitive aspects of discourse coherence in man and machine. Philosophy of mind and its consequences for empirical theories in cognitive science.

Logical approaches to knowledge representation; reasoning with incomplete information, non-monotonic reasoning, reasoning about action and change.


Logic programming, programming languages semantics.

**Dimiter Driankov**, Ph. D., Linköping 1989. Assistant professor (*universitetslektor*), logic and AI.

Reasoning under uncertainty, many-valued logics, approximate reasoning, fuzzy control & systems, autonomous agents.


Knowledge-based systems, knowledge acquisition, software development environments, software reuse.
Distributed systems, object-oriented programming, object-oriented analysis and design, operating systems.

Thomas Falk, Econ. Dr., Stockholm 1976. Adjunct professor in economics of information technology. Chalmers University of Technology. Several previous affiliations.
Management of information technology.

Per-Olof Fjällström, Ph. D., Stockholm 1985. Associate professor (universitetslektor), theoretical computer science. Previous affiliation KTH and IBM. Group leader, ACTLAB. Director of graduate study programme.
Computational geometry, analysis of algorithms, data structures.

Programming environments and languages, scientific computing, debugging tools, incremental compilation technology, compiler generation, compilers and development tools for parallel hardware.

Theories/methods on problem formulation, business process and activity analysis, IS design and evaluation; ISD methods and customization of CASE tools; Humanistic science traditions and qualitative research methods.


Programming languages and systems, programming methodology, program manipulation, partial evaluation.


Syntax, semantics, and pragmatics of natural languages; discourse analysis; argumentation theory; semiotics; philosophy of language.


Library science and systems, hypertext and -media, knowledge organization and information retrieval, citation analysis and bibliometrics, computer support for personal and cooperative activities, virtual environments.

Expert systems and artificial intelligence applications, database technology, human-computer interaction.


Artificial intelligence, Natural language processing, especially empirically based computational models of human-computer dialogues.

Mariam Kamkar, Ph. D., Linköping 1993. Assistant professor (universitetslektor), computer science.

Software maintenance, software testing and analysis, program debugging, program analysis, optimization in compilers, multiparadigm programming languages.

Krzysztof Kuchcinski, Ph. D., Gdansk 1984. Associate professor (docent, universitetslektor), computer systems. Group leader, CADLAB. Previous affiliation Technical Univ. of Gdansk, Poland.

Computer architecture, computer-aided design of digital systems, VLSI, design for testability.

Software engineering, real-time systems, industrial evaluation of elements of new software technology.


Human-Computer Interaction, usability issues in information systems development and software engineering, usability-oriented design support.


Artificial intelligence, knowledge representation, reactive systems, autonomous systems, system theory.


Logic programming, formal language theory, amalgamation of programming paradigms.
Simin Nadjm-Tehrani, Ph. D. Linköping 1994. Assistant professor (vik. universitetslektor) computer science.

Modelling and verification of embedded systems, temporal logic, real-time systems, logic programming.


Business modelling, strategy planning, activity based development, information systems development, maintenance management, application packages, information management.

Ulf Nilsson, Ph. D., Linköping 1992. Assistant professor (universitetslektor), computer science. Group leader, LOGPRO. Previous affiliation State University of New York at StonyBrook, USA.

Logic programming and deductive databases; Evaluation strategies for query processing; program transformation and abstract interpretation.


Inheritance, default reasoning, taxonomical reasoning, object-oriented systems.


Automated synthesis of digital systems, formal description of hardware, hardware/software co-design, design for testability, VLSI, computer architecture.

**Birger Rapp**, Econ. Dr., Stockholm 1974. Professor of economic information systems. Vice president at large for IFORS. Editorial (advisory) boards to EJOR, IJMSD, JORBEL and Omega. President of the Pronova Research and Development Board in Sweden.

Accounting, business control, agency theory, IT and organization, production, economics.

**Tore Risch**, Ph. D., Uppsala 1978. Professor of engineering databases. Group leader, EDSLAB. Previously at Uppsala University, IBM Almaden Research Lab. (San Jose, CA), Stanford Research Institute, Syntelligence Inc. (Sunnyvale, CA), HP Laboratories (Palo Alto, CA), and Stanford University.

Database support for engineering and scientific applications, e.g., object-oriented databases, heterogeneous data bases, active databases, and real-time databases.


Representation of knowledge with logic, reasoning about actions and change, cognitive robotics, autonomous agents.


Knowledge Management, Knowledge Engineering, Industrial Software Engineering, Quality Improvement Paradigm, Experimental Research Methods.


Programming theory, programming languages, debugging tools, compiling technology, information management, business process modelling, CSCW.
Åke Sivertun, Ph. D., Umeå 1993. Research associate (forskarassistent) at LIBLAB. Geographical Information Systems - GIS. Communication of complex data and linking multi disciplinary models in GIS. Research in environmental programs, programs for medical geography, physical planning and decision support.


**Guest Researchers Engaged in Graduate Study Program**

**Roland Bol, Ph. D., Amsterdam 1991.** Guest researcher.
Previous affiliation CWI, Amsterdam.
Logic programming, non-monotonic reasoning, deductive databases, process algebra.

**Brant Cheikes, Ph.D., Univ. of Pennsylvania, 1991.**
Guest researcher.
Natural language processing and cooperative dialogue; architectures for response-planning systems, esp. intelligent help systems; simulator-based training systems. Applications of knowledge-based systems technology for operator training in the process industry.

**Robert L. Glass, M. Sc., University of Wisconsin 1954.**
Software engineering: software quality, maintenance, design, technical communication, creativity. The importance of the application domain.

**Witold Lukaszewicz, Ph. D., Warsaw University 1979.**
Guest professor. On leave from the Institute of Informatics, Warsaw University, Poland.
Knowledge representation, non-monotonic reasoning, programming methodology.
**James M. Nyce, Ph. D., Brown 1987.** Guest professor, computer and information science. Previous affiliation Brown.

Work and knowledge (medicine and academia); tradition, innovation and technology; hypertext and visual resource development paths.

**Jukka Paakki, Ph. D., University of Helsinki, 1991.** Associate professor of computer science (University of Jyväskylä, Finland). Guest researcher.

Programming paradigms, language design and implementation, attribute grammars, logic programming.


Artificial intelligence in education and training, instructional planning, collaborative learning and intelligent learning environments.