MONASH UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

INTRODUCTION TO COMPUTER SCIENCE AND SCIENCE HONOURS 2004

http://www.csse.monash.edu.au/hons/

1 General Matters

Welcome to the Honours course in Computer Science! We hope that you will have a successful and enjoyable year in your course. If you are having any problems related to this course, or if you need course-related or general advice you should talk to the coordinator:

- Computer Science and Science Honours: Bernd Meyer
  (Rm 148, Building 75, Clayton, bernd.meyer@infotech.monash.edu.au)

Please be aware that the Digital Systems Honours Degreee has a slightly different structure which is described in an additional handout. In case of any questions regarding this degree you need to contact the coordinator:

- Digital Systems Honours: Nandita Bhattacharjee
  (Rm 189, Building 75, Clayton, Nandita.Bhattacharjee@infotech.monash.edu.au)

Other people you may need to deal with include

- Karen Fenwick, (G62, Building 75, Clayton, Karen.Fenwick@infotech.monash.edu.au): day to day Honours administration such as changes of unit registration.
- Susan Saunders (Rm 156, Building 75, Clayton, Susan Saunders@infotech.monash.edu.au): room keys, proximity cards etc.
- The technical services unit: If you have any trouble with the IT equipment mail to `tech@help.csse.monash.edu.au`.

Important Note: Please be aware that if you want to contact any of us by email you have to make sure that your mail originates from one of the Monash domains for your student account (`student.monash.edu, csse.monash.edu.au, sci.monash.edu.au, infotech.monash.edu.au`). Due to the amount of spam floolding the university mail system we need to use spam filters. As these systems sometimes generate false positives there is a chance that your mail will not arrive if you send it from accounts such as hotmail, gmx or yahoo.

Library Orientation

There will be a library orientation session at the start of the semester on **Thursday March 18, 10 am**. The focus of this session will be on using the library as a research resource and will include a hands-on session with electronic databases. Please meet Ms Sara Miranda at the Information Desk, Hargrave-Andrew Library; the session will be conducted in the IT training room.

Information resources

Information and handouts are available on the web at: `http://www.csse.monash.edu.au/hons/` where you will also find links to the subject homepages. **It is necessary that you read the email sent to your student account regularly as many important announcements are distributed in this way.**

2 Course structure

Each Honours student must undertake coursework units and a substantial individual research project which together add up to 48 points.

CSE417 Communication and Research Skills

All students must take the compulsory unit, CSE417 Communication and Research Skills. This unit is intended to improve the oral and written presentation skills of students and to teach skills required for the critical analysis of research. CSE417 will take place over both semesters. You will receive a separate handout for this unit in your first class. The first session takes place **11-12.30noon, Thursday 4 March, Room G55 (Seminar Room), Building 75**.

Seminars

School seminars are held regularly throughout the year (typically once a week). Information on CSSE seminars is available at: `http://www.csse.monash.edu.au/seminars/`.

We consider these to be part of CSE417 Communication and Research Skills. You are required to attend at least 5 seminars each semester and fill out a seminar evaluation sheet (available in separate handout and for download) for each seminar you attend and submit it immediately after the seminar to the Honours coordinator. Also, attendance at the interim and end of year honours project symposia is mandatory.

Coursework Units

The units offered this year to Computer Science students are detailed in the appendix and on the Web. A total of 24 coursework points must be taken (this does not count the Communications and Research Skills Training unit CSE 417, which is considered part of the research training). You must select 4 units (each counting 6 points) from the following list:

- CSE4601 - Advanced Topics in Intelligent Information Processing
- CSE4602 - Advanced Topics in Software Engineering
- CSE4603 - Advanced Topics in Algorithms and Complexity
- CSE4604 - Advanced Topics in Computational Languages
- CSE4605 - Advanced Topics in Computational Science
- CSE4606 - Advanced Topics in Computer Networking

- CSE4607 - Advanced Topics in Graphics and Visualisation
- CSE4608 - Advanced Topics in Computer Architecture and Systems
- CSE4610/11 - Individual Study Unit in Computer Science

Instead of choosing four CSE460X subjects from the list above you may choose only three and fill the remaining 6 points of coursework with a free elective. This can be a third year or fourth year subject or even a subject from another school or faculty. However, please note that the choice of such a free elective must be individually approved by the Honours coordinator and must be a reasonable addition to a Computer Science curriculum.

As you can see from the title these units cover rather broad topic areas. They are in fact flexible "framework" units in which you can specialise in different directions. Each of these framework units is worth 6 points for which you must elect one or two modules within the unit (some modules are worth 3 points, other ones six points).

More detailed descriptions of the CSE460X units and the modules within them is given in the appendix.

Modules typically comprise 36 or 24 hours of lectures over 12 weeks and include some practical work. Some modules are taught for a whole semester, while others are taught during only one half of a semester. Modules start dates are Week 1, Semester 1; Week 8, Semester 1; Week 1, Semester 2; Week 8 Semester 2.

The number of lecture hours is not necessarily the same for all 3 point modules or for all 6 point modules. The variation of lecture hours is balanced by, e.g., more/less reading or mor/less assignments, so that the overall workload of a six point unit will be independent of the module selection.

Assessment for each module may be based on assignments, an examination, or both, and will be clearly specified by the lecturer at the start of the module.

Unit and Module Selection

Note that you **cannot** directly enroll in the CSE460X units above without approval from the honours coordinator. This is a safeguard to ensure that your module selection is consistent with your unit selection.

Note that your enrollment with the faculty only concerns the framework units. Your selection of modules that you wish to count towards these units is done with the school.

On your day of enrollment (or before) pick up a **Honours Unit Selection Form** from the general office. This form clearly indicates which modules can be counted towards which units.

You must select four CSE 460X units. For each unit you must select modules that count for a total of 6 points. If you wish to substitute a free elective for one of the CSE460X units, you only need to select three CSE460X subjects and to additionally fill in the elective on the unit selection form.

Note that the CSE460X units are full year units.

After you have made your selection, you must have your **Honours Unit Selection Form** checked and approved by the coordinator who will also approve your unit selection form so that you can formally enroll into these units with the faculty.

You must also enroll into CSE 4650, which is the 24 point research component of the course.

If you later need to *change a unit* you can do so until the second week of Semester 1.

If you intend to *change a module* later in the year, you must formally notify your intention to do so by sending an email to Karen Fenwick (Karen.Fenwick@infotech.monash.edu.au). You will receive email approving (or not) the change. All changes must be approved. Under most circumstances only module changes that do not require a change in the unit selection can be approved.

Should you fail to formally notify the School of module changes or fail to get approval, marks for the original modules will be used to calculate your course work component.

Final Grade

The final grade (H1, H2A, H2B, H3 or fail) for the Honours course is computed by combining the project mark and coursework marks weighted in accordance with their point value. The coursework mark includes the CSE417 (Communication and Research Skills) unit marks and the best possible combination of modules/units that constitute 24 points. At most one free elective (non CSE460X) can be counted towards the final mark. In summary:

24 points of units	24 points
Honours thesis & CSE417	24 points
Total	48 points

Prizes

The name of the Dux of the year is inscribed on the Honours's board in the School of Computer Science and Software Engineering (Clayton) each year. Two prizes of A$500 each for the best students in the categories best coursework and best project have generously been donated by ManageSoft (`http://www.managesoft.com`).

3 Projects

The core of the Honours programme is an individual research project which is worth 20 points. Two important parts of the project work are written and verbal presentations of the project.

For many Honours students the Honours project is unlike anything they have done before. Sometimes it is hard to know what you should be doing and when and how you should be doing it. Here are some general guidelines. However, they are not applicable to all projects and your supervisor will be able to – and should – provide more project specific guidelines and goals.

The Research Project is designed to take about 500 hours for the average student. A Research Project may be concerned with theory, program development, hardware development, evaluating and improving on a new technique, analysing performance - in fact anything associated with computing which involves a reasonable amount of intellectual and practical effort. The student is expected to read the relevant literature and carefully analyse the problem posed, to formulate a solution or proposals for a solution, and where appropriate, to implement and prove, evaluate or test the validity of their results and proposals. The project solution will usually require creative and original thinking. Typically, a project is designed for a problem in some area associated with a research programme being carried out by a member of staff. The Research Project involves substantial

mentoring by a staff member, and is designed to teach research skills. These skills are particularly important if the student wishes to undertake a post-graduate research degree.

Project Registration Procedure

A list of projects will be handed out separately and is also available on the Web. It contains a brief outline of each project and the name of the supervisor. Supervisors can individually provide further information about a project and are willing to discuss what is involved.

During the first week of the semester there will be some project presentation session in which individual groups of supervisors present potential projects. Each such session will be dedicated to projects that are grouped around a common research area. The timetable for these sessions is available at `http://www.csse.monash.edu.au/ berndm/Honours/2004/ProjectIntros.html` and in hard copy on a separate handout. Make sure to attend these sessions if you have some interest in their topics (and even if you don't know what they are about), but be aware that not all projects are represented in these events.

If you have a project in mind that you would like to do you are encouraged to approach a relevant member of the academic staff to suggest this project. Most supervisors appreciate this sort of initiative and will be happy to supervise such a project if it is well-defined and in their area of research. However, students do not have a right to insist on their own project, and in the great majority of cases they will do projects from the published list. Students may not always be able to do the project of their choice, either due to the popularity of a project or because each staff member is restricted to supervising a maximum of three projects.

Project registration and allocation will be done in the first week of Semester 1. Please fill in the **Project Allocation Form** which you receive in the introductory session. Give at least 6 preferences in ranked order. **Your preferences must be supervised by at least 3 different supervisors.** Hand your completed form in to the Enquiries Office by the end of the first week of Semester 1. The project allocation will be announced by email as soon as possible after this time.

Evaluation of Projects

The School takes the evaluation of projects very seriously, as they are a substantial contribution to the student's final mark. The following are part of the project assessment.

NOTE: Some of these items are part of the assessment for CSE417 Communication and Research Skills; see CSE417 handout for details of assessment weightings.

1. A **research proposal** is due by 12 noon **28.4.04**. By this time the student is expected to have read sufficient literature to be able to form a fairly good plan of how to attack the chosen problem. Thus, this report typically contains a description of the project and plans for the solution of the associated problems. The report should show evidence of the thinking the student has done about the project, and possibly some initial experiments which indicate that the suggested approach seems plausible. A time-table accompanies the said plans. A submission must be made by the due date and handed in to the Enquiries Office. Re-submission will be requested for unsatisfactory proposal.

2. A **symposium** on **3.6.04** will be organised which all academic staff and Honours students will attend. Each student will be given approximately 10 minutes to describe their project

and what they have done on the project so far. A short question and comment time will follow. This seminar is intended to give experience in such presentations and to provide for input from other academic staff concerning your plan and approaches.

3. A first **Draft of a Literature Review** is due by 12 noon **11.6.04**. You have roughly six weeks for revision of this draft after which you must submit the **Final literature review** (Submission Deadline 28.7.04).

4. A first **Draft of the Thesis** is due **8.9.04**, 12 noon so that the supervisor can provide comments before the final report is submitted. (You should of course get drafts to your supervisors before this date.)

5. The **Final Report** is due in the first week of November on **Melbourne Cup Day, 2.11.04**. This is followed by a School Cup Day gathering to which Honours students are invited. The final report provides the basis of the project assessment and is examined by at least two staff members, one of whom is usually the supervisor of the project in question. It may be appropriate to demonstrate your project to your examiners.

   Only under exceptional circumstances will an extension of the thesis submission deadline be granted by the Honours coordinator.

6. The **Final Talks** will be scheduled on one or two days during the last week of October. during the week of **25.10.04-29.10.04** (exact dates to be advised). You will typically be given 20 minutes time for your presentation with 5 minutes time for questions and comments. Examiners take the seminar presentation and fielding of questions into account when assessing a project. It is compulsory for you to attend all final talks.

7. A *Project Web Site* must be maintained during the semester to keep supervisors, fellow students informed about your project and to impress the rest of the world. Its final version has to be available for marking one week after the final talks (**11.11.04**).

The final project grade is determined from the marks assigned by the two examiners, although their recommendations are sometimes changed by consensus to ensure that all projects are fairly marked. The examiners take into account each of the above, with emphasis given to the final report. If the two examiners' marks differ significantly, then a third or even fourth detailed examination may be called for. The final grades for the project and overall grade for the year are determined at a staff meeting after all components of the assessment have been marked. An external assessor will also independently examine selected projects and will be present at the staff meeting so as to ensure objective marking.

Most projects have a significant practical content, involving hardware and/or software development. It is crucial that by the time of the research proposal you have reached some agreement with your supervisor about the extent of this practical work. It is equally important that by the time you hand in your final report this practical work has been completed, as you are likely to lose marks for incomplete work. Your practical work will be judged for its quality in at least the following categories;

- correctness and completeness,

- sound, modular design,
- testing,
- ease of use,
- documentation.

Some projects will have a large theoretical component. Theoretical work will be judged for its quality in at least the following categories;

- formal correctness,
- problem understanding,
- innovative and independent problem solving,
- conceptual elegance and simplicity.

Guidelines for Carrying Out the Project

For most students, the hardest part of the Honours year is managing their time so as to work consistently on the project throughout the year amongst the short-term pressures of course work assignments and exams. Start your project **early** (that is, in March) and keep at it. Your project is worth almost half of your final marks for the year! Do not get bogged down spending a disproportionate amount of time on small course work assignments which are worth relatively little in your overall mark.

A rough timetable for your project should be:

- **1st week of semester**. Choose project and supervisor.
- **First 3 weeks of Semester 1**. Do background reading – typically 5 to 10 papers. Understand the problem and why it is important.
- **Weeks 4-6 of Semester 1**. Plan how to solve the problem. Write research proposal.
- **Rest of Semester 1** Undertake preliminary tests of your approach. Prepare interim presentation.
- **First 9 weeks of Semester 2**. Finish most of the implementation or proofs in your project work. First draft of your report.
- **Rest of Semester 2**. Write your report, and perform extra research suggested in the writing up stage.

The following hints may help your research and time management:

- You should have a regular weekly meeting time with your supervisor. Attend the meeting even if you have not completed anything.

- Keep a research diary where you write down your research ideas, progress on the project and a log of your meetings with your supervisor. This is a hurdle requirement for CSE417 Communication and Research Skills.
- Allocate 25% or 40% of your time to your project as appropriate. Regardless of course work requirements, spend at least one day a week on your project.
- Choose reasonable weekly and monthly goals in consultation with your supervisor.
- Read the background material critically – look for the underlying ideas, limitations and usefulness. Do not believe anything blindly just because it has been published.
- Start writing early. Do not leave it until the last weeks. Your research proposal and your literature review should form the basis of the first draft of your final report.
- Get help with your proposal, literature review, poster and final report:
  - your supervisor will help with structure and contents
  - you will receive help with your presentation in your CSE417 classes.
  - LLSU can provide help with ESL issues and scientific writing.
  - other students can help – form a self help group.
- Discuss your project with other Honours students. Communication is one of the keys to successful research.
- Practise your talks. In the final talk a **working** demonstration or video is often impressive.

4 Guidelines for the Final Report

The project report **must be typed** on A4 paper. It should typically be **no longer than 30 pages**, excluding the literature review, appendices and bibliography. Exceptions to this rule can be made if they are justified by the nature of the project. If you think your project constitutes such an exception you will have to discuss the case with your supervisor(s) and the coordinator and obtain their approval to exceed the page limit.

Draft printing must be kept to a minimum; use the previewers or WYSIWYG packages. A quota of pages produced on the laser printers may be imposed for each Honours student. Three copies of the report must be submitted if you have one supervisor, four if you have two supervisors. Photocopying of the copies can be arranged through the Enquiries Office.

It is important that the report contain a complete account of the work done. In general, the report should contain:

- a clear description of the problem you have tackled and why it is important,
- a description of how your project relates to related published work,
- the approach taken (with ample justification), and also methods used to solve the problem,
- the results or conclusions obtained, and

- if appropriate, a detailed description of the software and/or hardware you have implemented.
- if required an appendix that details how the deliverables produced in the project deviate from the deliverables spelt out in the research proposal (see documentation for CSE417) and why this is the case.

It is vital that the thesis contains a complete account of the work you have done: In particular, you should use an appendix to clarify your personal original contribution and to distinguish it from ideas and results that you have taken from the literature or that your supervisor has contributed. *Make clear what your own achievements and contributions are and how much time you spent on your project.* Your second reader will sometimes know your project only superficially, and your thesis is the best way for him or her to get to know it better.

With scientific writing, organisation and structure is half of the task, and so considerable effort should be invested in detailed outlines before any text is composed. Changing outlines is quick and easy; rewriting text is time consuming.

The supervisor will advise on all aspects of the preparation of the thesis, and will check through the draft at least once if received by the first draft deadline, but the student is reminded that it is not the supervisor's responsibility to write or re-write all or part of the work. Refer (with caution) to existing Honours theses of the School for an indication of the required format. Note that as a student you are being examined not only on research and organisation ability, but also on your ability to present and defend ideas. Conformity to conventions, both scientific and grammatical, is important.

A reasonable thesis structure is as follows:

- Title Page
- Declaration of Originality
- Thesis Abstract
- Acknowledgements
- Table of Contents
- List of Figures
- Introduction
- Main Body of thesis (Chapters or sections depend on topic)
- Conclusion and Further Research
- Appendix A: Revised Specification of Deliverables (if required)
- Appendix B: Clarification of Original Contribution (if required)
- Other appendices as required
- Bibliography

If you choose to use LaTeX, a suitable style file will be provided. The recommended font size is 11 point. Essential footnotes are normally placed at the foot of the page to which they refer. Number pages consecutively, including pages carrying diagrams, photographs, maps, etc. Diagrams should be computer drawn and included as postscript/latex graphic/etc files directly into the document, or at least photocopied onto the particular page. Photographs must be mounted with dry mounting tissue or spray adhesive, and where possible copied photographically as a whole page and included in the thesis in the normal manner. References must be referred to in the text, and listed in the bibliography following a standard and consistent format.

The "Declaration of Originality" must be on a separate page and contain the following wording:

> I <student name> declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.
>
> _____
> (student signature)
>
> _____
> Date (day, month and year)

The Abstract on a separate page should not exceed 500 words.

Appendices are not intended as a means to 'pad-out' a sparse thesis with peripheral material, or to circumvent the page limit in an 'obese' thesis. They serve as a repository for useful products of the research (e.g., documentation including installation of a program and a detailed example run of the program) which are not an integral part of the main body of the thesis. Where the raw data of a thesis cannot be extracted directly from the test figures and tables, it is essential that they be tabulated in an appendix. In short, appendices preserve valuable information which might otherwise be lost, but the thesis should be able to stand without them. Long, detailed program code should be put on a CD ROM or floppy disk in the back of the thesis, rather than listed in appendices.

In addition, student have to submit a copy of their thesis electronically to the coordinator (either PDF or HTML or Postscript).

Guidelines for the Research Proposal and Literature Review will be in the handout for CSE417 Research and Communication Skills.

5 Postgraduate Study

All Honours students should consider their potential and options for continuing into some form of research study. The Honours year can be seen against a number of backgrounds: employment advantages over 3 year degrees, ability to gather additional course material, even as a procrastination over career choices! But the original purpose of the Honours year is to provide training for students who wish to continue on to postgraduate study. This is still one of the main objectives of the degree,

and an understanding of this will doubtless help you to make sense of much of the course work you do get. If you are interested in postgraduate study, make your interests and desires known to your lecturers and project supervisor. They will be only to happy to help you gain additional insights and perspectives on what is to them a fascinating field of study. Not only will they enjoy your interest, but you may find it gives you the additional impetus to do well in what may be your final year of formal examinations. Good luck in those examinations!

Scholarships are available to support you. They start at about $15,000 per annum (tax-free), and can be supplemented to higher figures in particular circumstances. There are a number of different types of scholarship, the main ones (roughly in order of prestige and amount) being:

- Australian Postgraduate Awards (APA).
- Monash Graduate Scholarships
- Faculty of Information Technology Scholarships
- School of Computer Science and Software Engineering Scholarships

Students intending to apply for scholarships are urged to talk to the Postgraduate Coordinators (Dr. David Squire and Dr. Andrew Paplinski). There are a number of options available to those who miss out on the very competitive APA and MGS awards. Competition for all scholarships is fierce. Usually only students with an H1 grade for Honours are successful in obtaining scholarships.

Further announcements about Postgraduate Study will be made later in the year. Also see the Postgraduate Handbook on the School WWW entry.

6 Appendix A: Subjects offered

Note that the framework units as such have no further prerequisites if you have been admitted to the BCS Honours programme. However, individual modules that you wish to count towards these units may have additional prerequisites. Please check in the module descriptions in the appendix.

CSE4601 - Advanced Topics in Intelligent Information Processing

Methods from Artificial Intelligence (AI) form the basis for many advanced information systems. These techniques address problems that are difficult to solve or not efficiently solvable with conventional techniques. Building on the undergraduate curriculum this unit introduces the student to advanced AI methods and their applications in information systems. Within the framework of this unit, the student can choose between a variety of modules in the broad area of Intelligent Information Systems. Most modules relate directly to the school's research strengths and are taught by active researchers in the respective fields. Research fields covered include:

- Machine Learning
- Data Mining and Knowledge Discovery
- Automated Reasoning
- Knowledge-based Systems Search,
- Constraint Solving and Optimization
- Natural Language Processing
- Computer Vision and Pattern Recognition

Some of these topics may not be offered in every year.

CSE4602 - Advanced Topics in Software Engineering

Software engineering is concerned with all aspects of effectively building reliable software systems that satisfy the requirement. It addresses the entire software life cycle including requirement analysis and specification, design, construction, testing, and operation and maintenance.

The modules in the framework of this unit cover advanced issue in software engineering, particularly the use of formal methods, ie.

- Mathematical concepts behind formal methods,
- Executable specifications,
- Formal specification formalisms and their use in the software engineering process,
- Verification and testing.

CSE4603 - Advanced Topics in Algorithms and Complexity

Algorithms are the most fundamental area for all aspects of computer science and software engineering. Discrete structures, such as those treated in graph theory, set theory, combinatorics and symbolic logic form the mathematical underpinning of the study of algorithms. As well-designed algorithms and data structures are essential for the good performance of an information system, an in-depth understanding of the theoretical properties of algorithms is essential for any computer scientist. As importantly, the theoretical investigation of algorithms leads to a deeper understanding of problem structures and classes of problems and the knowledge of a large variety of algorithm types enables the designer to approach a new problem from different angles.

Within the framework of this unit, the student can choose between a variety of specialisation modules in Algorithms and Discrete Structures. Most modules relate directly to the school's research strengths and are taught by active researchers in the respective fields. Research fields covered include:

- Computability and Complexity
- Automata Theory
- Advanced Analysis and Design of Algorithms
- Parallel and Distributed Algorithms
- Numerical Algorithms
- Cryptographic algorithms
- Spatial/geometric algorithms

Some of these topics may not be offered in every year.

CSE4604 - Advanced Topics in Computational Languages

Advanced working knowledge of programming languages is central to most activities in computer science. As students can expect to use many different languages and types of languages in their professional work, they should acquire knowledge of more than a single paradigm.

Modules in the framework of this unit

- introduce the student to different programming paradigms (such as logic programming, functional programming, agent-based programming)
- introduce specialized modelling/programming languages for particular types of problems (such as mathematical modelling languages, constraint languages and parallel languages)
- discuss formal concepts underlying programming languages, in particular (a) programming language semantics (b) type systems and calculi,
- discuss advanced issues in the implementation of programming languages

Some of these topics may not be offered in every year.

CSE4605 - Advanced Topics in Computational Science

All sciences are increasingly relying on computational support and the growth of many branches of science has only become possible due to the availability of efficient computational methods. The common basis of such methods are numerical methods and high performance computing.

Under the umbrella of this unit, the student can specialize in a particular areas of computational science by choosing from different modules including:

- Numerical Methods
- High Performance and Parallel Computing
- Optimization and Operations Research
- Bioinformatics
- Simulation, Visualization and Modelling

Some of these topics may not be offered in every year.

CSE4606 - Advanced Topics in Computer Networking

With the rapid growth of the internet and increasing use of company-internal networks, network-oriented computing has become a central field in the discipline. Within the framework of this unit, the student can choose between several modules which cover different advanced areas of network computing. Most of these modules relate directly to the school's research strength and are taught by active researchers in the respective fields. Topics covered include:

- Digital Communication Technologies: Local area networks; metropolitan area networks; satellite networks; ISDN; modem techniques; digital networks
- Network security
- Internet Protocols
- Advanced Network design and management
- Compression and Coding Methods
- Quality of Service

Some of these topics may not be offered in every year.

CSE4607 - Advanced Topics in Graphics and Visualisation

This unit covers advanced topics in computer graphics and visual interfaces. Within the framework of this unit, the student can choose between a variety of modules relating to these sub-fields. The topics relate directly to the school's research strength and are taught by active researchers in the respective fields. The fields covered include:

- Advanced image synthesis, including: polygonal rendering; local and global illumination models; hidden surface removal algorithms; parametric curve and surface representations; texturing; sampling and aliasing theory; and advanced lighting models.
- Advanced animation and modelling techniques, including: specification of models (mountain landscapes, plants, animals), their movements and high-level behaviour. Various means of giving "Artificial Life" to what are essentially sets of numbers are examined: dynamical systems including cellular automata and reaction-diffusion systems; explicitly, implicitly and aesthetically-directed genetic algorithms; virtual worlds and ecosystems; physical simulation.

Some of these topics may not be offered in every year.

CSE4608 - Advanced Topics in Computer Architecture and Systems

This unit covers topics in hardware architecture ranging from the gate level to processors and full computer architecture. Topics include

- Gate-level architecture
- VLSI design
- Hardware description languages and their application
- Hardware design specifications and methodology
- Software tools and packages for design, specification and
- Selected algorithms for digital design: Multipliers. Random Number Generators.
- FPGA, ASIC, etc.
- Machine organization
- Memory structures
- Architectures for parallel processing

Within the framework of this unit, students can select individual modules to specialise in a particular domain, such as VLSI design or parallel architectures. Some of these topics may not be offered in every year.

CSE4610 - Individual Study Unit in Computer Science / 0 point

This unit allows the student to study additional material and/or related fields pertaining to the topic of his/her chosen research project. Its contents is therefore individually defined. Please note that this is a 0 point unit, so you cannot count it towards fulfilling your degree requirements. However, the unit will appear on your transcripts so that your additional studies are documented.

CSE4611 - Individual Study Unit in Computer Science / 6 point

This unit covers advanced current research topics in computer, new emerging trends and research directions that are not covered in any other honours unit. Enrollment requires individual approval and it may not be offered in every year.

CSE4650 - Individual Honours Research Project

This subject introduces the student to independent research. Most projects are software-oriented, although some projects may be purely theoretical and others may involve hardware work.

A research project covers the whole process from initial problem analysis in a current research topic of computer science, literature study and evaluation of existing research and proposal of a research plan to carrying out the proposed research and presenting it in written and oral form. Where appropriate it includes the development of software (or hardware), from analysis through design to implementation and testing and documentation.

The project is conducted by the student in close cooperation with one or several staff members. The staff member will initially lead the project, help to formulate the initial research question and guide the student throughout the project. The staff member will arrange meetings with the student (typically weekly) in which intermediate results are reported and analysed and further directions for the project are decided on. The student is expected to read the relevant literature and carefully analyse the problem posed, to formulate a solution or proposals for a solution, and where appropriate, to implement and prove, evaluate or test the validity of their results and proposals.

The formal research skills training comprises weekly lectures as well as supervised literature study. Individual consultation is offered additionally for the improvement of presentation skills. Students also attend and evaluate regular school research seminars.

The research project is complemented by formal research training which is designed to improve the oral and written presentation skills and to teach the skills required for a critical analysis of current research. This component comprises lectures and seminars on presentation structuring, writing and editing, literature study, research methods, argument analysis and analysis of experiments and design and delivery of oral presentations.

ASSESSMENT: Project Evaluation by Supervisors and Additional Examiner(s) based on Deliverables, Final Report and Final Presentation (85%), Written Research Skills Assignment (5%), Presentations (10%): consisting of Initial Research Proposal (20%), First Seminar Presentation (mid-year interim presentation) (Hurdle), Literature Review (30%), Project Presentation as web site and/or poster (20%), Final Seminar Presentation (30%).

7 Appendix B: Modules offered

Note that not all modules are offered in every year. For details of which modules are offered in 2003 and which module may be counted towards which framework unit, please consult the *Honours Module Selection Form.*

Module 417: Communication and Research Skills

Harriet Searcy, Kevin Korb, Linda McIver, David Squire and Honours Coordinators,

This module is compulsory and counts as part of your research training component unit CSE4650. *You cannot count this module towards any other framework unit CSE460X.* The module covers three main areas: research skills; technical writing; and technical presentations. You will learn to: research, deliver and evaluate professional technical presentations; write a literature review; structure and write a thesis. (See also description of CSE 4650 above).

Module 450: Procedural Modelling, Animation and Artificial Life in Computer Graphics

3 points
Alan Dorin

This course covers the procedural specification of models for animation, their basic movements and high-level behaviour. Various means of giving "Artificial Life" to what are essentially sets of numbers are examined. These are utilized in the subject's assignment questions which provide an opportunity to gain practical experience in the production of models for computer animation, as well as in the rendering and display of these models.

Topics covered include algorithms for modelling: mountain landscapes; trees, shrubs and vines; smoke, clouds and rain; animal group behaviour e.g. ants, wasps, birds, sheep, fish etc.; animal movement animal morphology cellular growth patterns

Issues/areas studied include: emergence of complex global outcomes from local interactions; dynamical systems including cellular automata and reaction-diffusion systems; explicitly, implicitly and aesthetically-directed genetic algorithms; virtual worlds and ecosystems; physical simulation;

Recommended Reading:

Watt, A & M Watt (1992), "Advanced Animation and Rendering Techniques: theory and practice" ACM Pres, New York NY

Additional reading and research is required and will be indicated in the lecture notes during the course.

Prerequisities:

Knowledge of basic computer graphics principles including: Raster display techniques: line-drawing algorithms, rasterization algorithms Vectors, matrices & transformations in 2D and 3D computer graphics Principles of 3D projection (e.g. orthogonal and perspective projections); Principles of shading (e.g. flat, Gouraud and Phong shading); Elementary calculus

Familiarity with OpenGL or similar graphics library; Programming under UNIX in C/C++ or Java at 3rd year computer science level A thorough understanding of algorithms and data structures

This prerequisite knowledge can be gained in CSE3313 (Computer graphics), MAT 1811/MAT 1812 (mathematics), a major individual project in C/C++ or Java.

Module 451: Advanced Image Synthesis

3 points
Jon McCormack

This course covers advanced topics in computer graphics and image synthesis. It aims to provide a solid introduction to the advanced theory and practice of modern computer graphics, including: polygonal rendering; local and global illumination models; hidden surface removal algorithms; parametric curve and surface representations; texturing; sampling and aliasing theory; and advanced lighting models. Successful completion of the course will give the student the necessary skills to undertake further research topics in computer graphics and the necessary background for commercial applications in areas such as computer games, visualization, and SFX for film and television.

Recommended Reading:

Watt, A & M Watt (1992), Advanced Animation and Rendering Techniques: theory and practice, ACM Pres, New York NY, 1992.

Additionally research publications will be handed out in the lectures

Prerequisities:

Foundations of Computer Graphics as taught in CSE3113 or equivalent: basic mathematics for computer graphics: linear algebra, vector calculus, geometry, numeric methods. ; basic line, curve and surface representations and drawing algorithms ; standard 2 & 3 dimensional graphics transformations and projections (perspective, orthographic) ; 2 & 3 dimensional clipping algorithms ; homogeneous coordinate systems and matrix representations ; basic OpenGL and GLUT programming ; basic hidden surface and line removal algorithms ; Gouraud and Phong shading algorithms.

Module 452: Cryptography I

3 points
Graham Farr

The subject matter falls naturally into two parts: Secret-key and public-key cryptosystems.

1. secret-key cryptosystems elementary systems: simple substitution, polyalphabetic substitution, Vigenere cypher, one-time pad, local transposition; basic principles of building and combining such systems: substitution, transposition, composition, confusion, diffusion; a modern secret-key cryptosystem such as the Data Encryption Standard: outline and main components;

2. one-way functions and public-key cryptosystems, principles of one-way functions, trapdoor functions, and public-key cryptosystems in general; relevant notions from computational complexity; some necessary mathematics, especially number theory: modular arithmetic, Euler totient,

inverses mod n, primitive roots, modular exponentiation, discrete log, extended Euclidean algorithm, factorisation; key exchange, Diffie-Hellman scheme, Shamir scheme; Rivest-Shamir-Adleman, Merkle-Hellman and ElGamal public-key cryptosystems; comparison of secret- and public-key cryptography.

Recommended Reading:
Bruce Schneier, Applied Cryptography: Protocols, Algorithms and Source Code in C (2nd edn.), Wiley, New York, 1996.

Prerequisities:
Basic ideas of complexity (as taught in CSE3305); 12 points of mathematics

Prohibitions:
CPE5001

Module 453: Cryptography II

3 points
Graham Farr

The module naturally falls into two parts:

1. Secret-key cryptosystems: information theory: information, entropy, sources, the Asymptotic Equipartition Principle, statistical structure of language, key and message equivocation, unicity distance, perfect secrecy.

2. Cryptographic protocols — mainly, using and combining cryptographic operations to do more than just straight encryption and decryption: authentication and digital signatures, using both secret- and public-key methods; further applications of one-way functions and public-key cryptosystems, such as bit-commitment and relatives; more advanced protocols, e.g. electronic cash, interactive proofs.

Recommended Reading:
Bruce Schneier, Applied Cryptography: Protocols, Algorithms and Source Code in C (2nd edn.), Wiley, New York, 1996.
Dominic Welsh, Codes and Cryptography, Oxford University Press, 1988.

Prerequisities:
Module "Cryptography-I"

Prohibitions:
CPE5001

Module 454: Data Mining and MML

3 points

Lloyd Allison

The subject includes: Elementary information theory (including noiseless coding and compression); inductive inference and prediction; data modelling and data mining; introduction to Minimum Message Length (MML) inference; clustering, mixture modelling and unsupervised classification; supervised classification and decision trees and related structures. Typical applications will be described.

Recommended Reading:

- C. S. Wallace and M. P. Georgeff. A General Objective for Inductive Inference. [TR 32], Dept. Computer Science, Monash University, March 1983.
- J. J. Oliver and D. J. Hand, Introduction to Minimum Encoding Inference, [TR 4-94] Dept. Stats. Open Univ. and also TR 94/205 Dept. Comp. Sci. Monash Univ.
- J. J. Oliver and R. A. Baxter, MML and Bayesianism: Similarities and Differences, [TR 94/206]
- R. A. Baxter and J. J. Oliver, MDL and MML: Similarities and Differences, [TR 94/207]
- The Computer Journal, special issue 42(4), 1999, includes:C. S. Wallace & D. L. Dowe, Minimum Message Length and Kolmogorov Complexity. pp270-283

Other relevant material will be published on the modules web site.

Prerequisities:

- Knowledge of Differentiation and integration of polynomials, exp(x), log(x) and combinations of these. Integration by parts. Evaluation of the determinant of 2x2 and 3x3 matrices. Probability theory: Prior-, joint-, conditional- and posterior-probability. Binomial distribution. Probability density. Normal distribution. Elementary information theory and relation to (lossless) coding. The multi-state and normal distributions, and their maximum likelihood and minimum message length estimators Encoding of mixture models and of decision trees.
- Knowledge of Numerical accuracy, rounding error, simple methods of solving f(x)=0 and of integration, Search and optimization methods, greedy search, look-ahead, exhaustive search, stochastic methods.
- Good ability with Unix operating systems
- programming in more than one programming language, e.g. C/C++ (gcc), Java, Perl

Knowledge of these issues can be gained in CSE 2304, CSE 3305 and the 2nd and 3rd year CSE core units.

Module 455: Minimum Message Length

3 points
David Dowe

The subject includes topics:

Foundations of inductive inference from (algorithmic) information theory; intermediate to advanced Minimum Message Length (MML) inference; details of Fisher information and uncertainty regions; angular/circular models (von Mises, Wrapped Normal or trigonometric); Poisson distribution; MML of specific models such as decision graphs, hidden Markov models (or HMMs, also known as probabilistic finite state automata, or PFSAs), linear and polynomial regression, causal models, Bayesian nets, time series, sequences, segmentation, trends; probabilistic prediction and Kullback-Leibler distance. Statistical invariance, statistical consistency. Data mining. Additional models may include factor analysis and additional theory may include the Neyman-Scott problem. Might also manage to fit in some or all of: polygon modelling; DNA pattern discovery and alignment, evolutionary trees; Lempel-Ziv text compression, C.S. Wallace improvement (1989, 1996), approximate repeats; HMMs (PFSAs) in mixture modelling; Markov fields, images.

Applications to be considered may include: models of protein folding and protein structure prediction, bushfire prediction, text and image analysis, DNA alignment and the human genome project, authorship identification for texts, etc. Further typical applications may be described. The exact composition of topics may vary.

Recommended Reading:

C.S. Wallace and D.L. Dowe (1999), "Minimum Message Length and Kolmogorov complexity", Comp. J., Vol. 42, No. 4, pp270-283.

Prerequisities:

Prerequisite knowledge:

- as taught in Module "Data Mining": A knowledge of the material in this module is strongly recommended.
- as in CSE3305 Formal Methods II: Probability theory: Prior-, joint-, conditional- and posterior-probability. Elementary information theory and relation to (lossless) coding. Binomial distribution. Probability density. Normal distribution.
- as in CSE2304 Algorithms and Data Structures: Numerical accuracy, rounding error, simple methods of solving f(x)=0 and of integration. Search and optimization methods, greedy search, look-ahead, exhaustive search, stochastic methods.
- as in CSE 2nd- and 3rd-year core subjects: Good ability with Unix operating systems, and with programming in more than one programming language, e.g. C/C++ (gcc), Java, Perl.
- Mathematics: Differentiation and integration of polynomials, exp(x), log(x) and combinations of these. Evaluation of the determinant of 2x2 and 3x3 matrices. A knowledge of classical statistics (e.g., maximum likelihood) and Bayesian statistics would be advantageous.

Module 456: Pattern Recognition

3 points
Sid Ray

Pattern Recognition deals with the study of theory and techniques used for the design of automatic devices capable of performing recognition of spatial and temporal patterns. Some of its well-researched application areas are: Recognition of handwritten and machine-printed characters, Automatic Speech Recognition and Speaker Identification, Analysis of Remotely Sensed Data, and Medical Diagnosis.

This course on Pattern Recognition covers discussion on Mathematical,Statistical and Fuzzy Set Theoretic methods of pattern recognition. Topics include:

- Introduction to PR Concepts and Methodologies;
- Statistical and Mathematical Preliminaries;
- Pattern Classification Methods including Bayesian Classification Rule, Linear Discriminant Functions, Minimum Distance Classification, Nearest Neighbor Rule, and Sequential Classification; Unsupervised Learning and Clustering; Feature Selection Methods including Mahalanobis Distance-Based Feature Evaluation;
- Fuzzy Sets and Their Application in Pattern Classification and Clustering.

Recommended Reading:

- J T Tou and R. C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1974.
- R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.
- K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.
- P. A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall International, 1982.
- A. Kandel, Fuzzy Mathematical Techniques with Applications, Addison-Wesley, 1986.
- G J Klir and B Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, 1995.

In addition, selected research papers will be referenced and discussed throughout the course.

Prerequisities:
Knowledge of: Introductory Probability Theory; Basic Satistics; Introductory Matrix Algebra.

Module 457: Data and Image Compression

3 points
Peter Tischer

In the handling of information it can be important to store the information in the most compact manner. This is known as data compression but is also sometimes called lossless data compression. In some circumstances we are prepared to discard information that is not considered important to reduce the storage requirements of the data. This is called lossy compression. The module deals with both lossless and lossy data for general data and for image and audio data. The course discusses a general approach for carrying out both lossy and lossless compression and concentrates mainly on image data. Existing methods as incorporated in international standards are examined as well as the theory necessary to understand state-of-the-art algorithms. In assignment work students will have the opportunity to implement compression methods they have developed themselves.

Topics include:

- pattern-based universal data compression techniques like Lempel-Ziv methods,
- the Langdon and Rissanen paradigm: data compression as modelling followed by coding,
- basic information theory (ideal code length, entropy, optimal coding: Huffman codes, arithmetic codes, Rice codes)
- universal modelling techniques (modeling intersymbol statistical dependence via higher order Markov models, modeling non-stationarity of data)
- compression of binary images (run length coding techniques, Group 3 and Group 4 fax algorithms, context-based coding, the JBIG and JBIG2 international standards)
- lossless compression of grey-scale images (prediction of pixel data, encoding of prediction errors)
- lossy compression of greyscale images (quantization, lossy DPCM coding)
- progressive coding
- transform coding (general case, Discrete Cosine Transform based lossy coding, the JPEG standard)
- compression of audio-visual sequences (MPEG1, MPEG2 and MPEG4 international standard)
- subband coding, wavelet coding (the JPEG2000 international standard)
- vector quantization coding
- relationship between data compression and Minimal Message Length Inductive Inference

Module 458: Bayesian Models

3 points
David Albrecht

In a wide variety of areas, including medical diagnosis, business investments, oil exploration, and weather prediction, people develop models to assist them in making rational decisions under uncertainty. This module will provide an introduction to Bayesian models and how they can be used in the decision making process. Topics include: Decision flow diagrams, extensive form of decision analysis, strategies, normal form of decision analysis, utility theory, probability theory, Bayesian Networks, Dynamic Bayesian Models, inference methods in Bayesian models, building Bayesian Networks, knowledge elicitation, approximation methods, Decision Networks, Dynamic Decision Networks, Markov Decision Processes, and Reinforcement Learning.

The exact composition of these topics and their relative weights in the schedule may vary.

Recommended Reading:

Russel, S, and P. Norvig, "Artificial Intelligence: A Modern Approach", Prentice Hall, Inc., 1995

Module 459: Causal Discovery

3 points
Kevin Korb

Causal discovery aims to develop algorithms to learn the structure of causal processes from observation. This goes to the heart of the long-standing dispute over whether we can learn causal relations from observed correlations. We will start with causal modeling techniques introduced in the early 20th century by Sewall Wright for dealing with linear causal structure. We will briefly review developments throughout the last century dealing with linear models and their causal interpretation, including structural equation modeling in economics.

In the late 1980s graphical models – Bayesian networks – became popular in AI for representing and reasoning with probabilities. In order to overcome the "knowledge bottleneck", researchers quickly turned to the problem of the machine learning of Bayesian networks from data. The techniques discovered are natural extensions of the linear modeling above. We will examine the main developments: The Verma-Pearl Conditional Independence Algorithm (1990), Tetrad II's PC Algorithm (1993), Cooper-Herskovits's Bayesian K2 (1991), Heckerman and Geiger's BDe/BGe (1995), Causal Discovery via MML (1996).

We will consider the question whether Bayesian networks are properly understood as fundamentally causal or simply probabilistic (i.e., correlation vs cause, again).

We also look at closely related questions, such as: learning probabilities from data; learning with incomplete data; Monte Carlo methods for automated learning; expectation maximization (EM) methods; evaluating machine learning methods.

The exact composition of these topics and their relative weights in the schedule may vary.

Recommended Reading:

- J Pearl (2000) Causality.
- R Neapolitan (2003) Learning Bayesian networks.
- Michael I. Jordan (Ed.) (1998) Learning in graphical models. MIT.
- Peter Spirtes, Clark Glymour, Richard Scheines (2000) Causation, prediction, and search. Springer Verlag.

Research papers will be referenced throughout the subject.

Prerequisites:
The module "Bayesian Models" is recommended.

Module 460: Optimisation and Constraint Solving

3 points
Bernd Meyer

Optimisation and constrained solving problems are of extreme importance in industrial applications, such as timetabling, resource allocation, airline scheduling or fleet coordination. Unfortunately almost all of these problems are computationally hard and therefore specialized techniques are required to handle them. This course discusses the various paradigms and methods that can be used to solve (constraint) optimisation problems. Theoretical foundations and application areas of the various methods will be presented and hands-on experience will be provided in practical assignments in which this knowledge has to be applied to concrete problems.

Topics include:

- Characterisation and examples of optimisation problems (Continuous vs. discrete problems, Unconstrained vs. constrained optimisation problems),
- Computational complexity of optimisation and satisfaction problems (Intractable problems: NP-complete/hard problems, Cook's theorem),
- Techniques for Handling Intractable Problems (Heuristics, Stochastic, Probabilistic Algorithms)
- Basics of Constraint satisfaction problems,
- Constraint Solving, Gauss-Method,
- Linear programming, Simplex method,
- Integer and Mixed Integer Programming (Network flow, branch&bound methods, Modelling with integer programming),
- Non-linear constraint optimisation (Euler-Lagrange Multiplier Methods, Kuhn-Tucker Conditions, Quadratic Programming),

- Stochastic search methods and their origins in nature (Boltzman machines, Simulated annealing and its variants, Evolutionary methods, evolution strategies, Genetic algorithms and genetic programming, Ant colony optimisation, Artificial immunes systems, Neural networks applications in optimisation, mean-field theory)
- Tabu search.
- Industrial Applications of Constraint Optimisation Techniques.

The exact composition of these topics and their relative weights in the schedule may vary

Recommended Reading:
Recommended books covering the basics for this subject are:

- Introduction to Mathematical Programming. Wayne L. Winston. Duxbury Press, 1995.
- Operations Research
- Wayne L. Winston. Duxbury Press, 1995.
- Constraint Programming-An Introduction. Kim Marriott and Peter Stuckey. MIT Press, 1998.
- New Ideas in Optimisation. David Corne, Marco Dorigo and Frank Glover (Eds.). McGraw Hill, 2000.

In addition to this, selected current research papers will be referenced in the lectures.

Prerequisities:
knowledge of basic calculus and matrix algebra

Prohibitions:
ETC4480

Module 461: Constraint Programming & Modelling

3 points
Kim Marriott

Constrained optimisation and satisfaction problems are important in many industrial applications, such as timetabling, resource allocation, airline scheduling or fleet coordination. Unfortunately almost all of these problems are computationally hard and therefore specialized techniques are required to handle them. There are two aspects to solving such problems. The first is the techniques used to solve the problem. This is covered in the co-requisite module Optimization and Constraint Solving. The second aspect is how to model and actually solve these problems in practice. This is the subject matter of this module. It covers the three main approaches: mathematical modelling language, constraint programming and object oriented constraint solving toolkits and their associated modelling methodology. Mathematical modelling languages will be presented as an example of a

domain specific programming language designed for use by non-programmers. Hands-on experience will be provided through practical assignments.

Topics include: - Mathematical modelling languages (For linear and integer programming, For constraint programming) - Constraint programming (Node and arc consistency, bounds consistency, generalized consistency, Backtracking search, Constraint logic programming (CLP), Constraint handling rules, Other constraint programming languages) - Object-oriented constraint solving toolkits (For linear and integer programming, For constraint programming) - Examples of Constrained Optimisation and Satisfaction problems (Resource scheduling, Health care, Bio informatics, Computer graphics)

The exact composition of these topics and their relative weights in the programme may vary

Recommended Reading:

Recommended text books for this subject are:

- AMPL: A Modeling Language for Mathematical Programming. Robert Fourer, David M. Gay, and Brian W. Kernighan. Duxbury Press / Brooks/Cole Publishing Company, 1993
- The OPL Optimization Programming Language. Pascal Van Hentenryck. MIT Press, 1999.
- Constraint Programming-An Introduction. Kim Marriott and Peter Stuckey. MIT Press, 1998.

In addition to this, selected research papers will be referenced in the lectures.

Corequisities:

Module "Optimization and Constraint Solving"

Module 462: Parallel Architectures

3 points
Ron Pose

The module includes the following topics: Parallel architectures; Bus-based shared memory machines; Massively parallel machines; Vector machines; Cluster computing; and Special purpose machines.

Recommended Reading:

Reading material including research papers, programming manuals and system specifications, will be distributed electronically or in hardcopy.

Prerequisities:

CSE2302 and CSE2/3324.

Prerequisite knowledge: types of parallelism within a computer architecture; processes, scheduling, inter-process communication; and experience with multiple programming languages.

Prohibitions:
CSE4333

Module 463: Parallel Programming

3 points
Trevor Dix

The module includes the following topics: Semaphores; shared memory vs. message passing; Linda paradigm; Data parallel programming; cluster programming.

Recommended Reading:
Reading material including research papers, programming manuals and system specifications, will be distributed electronically or in hardcopy.
Reference Material:

- G.R. Andrews: Foundations of Multithreaded, Parallel and Distributed Programming, Addison-Wesley, 2000.
- I.T. Foster: Designing and Building Parallel Programs, Addison-Wesley, 1995.
- M. Maekawa, A.E. Oldehoeft, R.R. Oldehoeft: Operating Systems Advanced Concepts, Benjamin/Cummings, 1987.

Prohibitions:
CSE4333

Prerequisities:
CSE2302 and CSE2/3324.
Prerequisite knowledge: types of parallelism within a computer architecture; processes, scheduling, inter-process communication; and experience with multiple programming languages.

Module 464: Information Security I

3 points
Tony Kerr

This units provides a broad introduction to security issues in information systems. The first part of the unit covers general issues in infomration security, such as physical security; network security; software security; contingency planning; legal issues; management issues.

Module 466: Natural Language Techniques in Information Retrieval

3 points
Ingrid Zukerman

The growth in popularity of the Internet and of large in-house knowledge repositories highlights the importance of developing machinery for accessing these resources. Natural Language Processing (NLP) and Information Retrieval (IR) are two essential tools for performing this task. This course discusses various NLP and IR techniques which support the access and retrieval of information from large knowledge repositories. Theoretical foundations and application areas of the various methods will be presented and hands-on experience will be provided in practical assignments in which this knowledge will be applied to concrete problems.

Topics include:

- Deterministic and stochastic grammars
- Syntactic analysis, Parsing algorithms
- Semantic analysis (Logical forms, Ambiguity resolution, Applied strategies)
- Corpus-based methods (Part-of-speech tagging, Probabilistic grammars)
- Pragmatics (Plan recognition)
- Information retrieval: indexing and search (document-based indexing, full-text indexing)

The exact composition of these topics and their relative weights in the schedule may vary.

Recommended Reading:

Recommended books covering the basics for this subject are:

- Natural Language Understanding – James Allen, 2nd edition, Benjamin Cummings Publishing Company, 1995
- Speech and Language Processing – Daniel Jurafsky and James H. Martin, Prentice Hall, 2000
- Foundations of Statistical Natural Language Processing – Chris Manning and Hinrich Schutze, MIT Press, 1999
- Modern Information Retrieval – R. Baeza-Yates, B. Ribeiro-Neto, Addison Wesley, 1999

Module 467: Epistemology of Computer Simulation

3 points
Kevin Korb, Aland Dorin, Jon McCormack, Bernd Meyer

How can computer simulations tell us anything about the world? What preconditions need to be met before they can? Is there anything different about a computer model of a physical system and a scientific theory about that system? Can one be true while the other is false? We will explore possible answers to these questions offered in various contexts from mundane to strange. We will look at a number of case studies, including the infamous pronouncements of the Club of Rome in the 1970s about impending environmental and population disaster, which may well turn out to be correct. We will also consider: computer network simulation; shopping market queuing simulation;

astronomical simulations; ethological and ecological simulation; artificial life forms and evolutionary ethics.

This will be a colloquium-style subject in which one or two papers will be read and discussed per week. Assessment will be by a short research paper assessed by one of the subject leaders.

Module 4213: Formal Methods in Software Engineering

6 points

For a description see CSE4213 in the handbook or courseware web pages.

Module 4431: System validation & verification, quality and standard

6 points

For a description see CSE4431 in the handbook or courseware web pages.

Module 4882: Digital Communication Technology

6 points

For a description see CSE4882 in the handbook or courseware web pages.

Module 4884: Network Design and Management

6 points

For a description see CSE4884 in the handbook or courseware web pages.

Module 4891: Public Telecommunications Networks

6 points

For a description see CSE4891 in the handbook or courseware web pages.

Module 4892: Information Security

6 points

For a description see CSE4892 in the handbook or courseware web pages.

Module 5301: Neuro-Fuzzy Computing

6 points

For a description see CSE5301 in the handbook or courseware web pages.

Module 5302: Digital Video Coding and Compression

6 points

For a description see CSE5302 in the handbook or courseware web pages.

Module 5312: Advanced Digital Design

6 points

For a description see CSE5312 in the handbook or courseware web pages.

Module 5803: Advanced Internet Protocols and Applications

6 points

For a description see CSE5803 in the handbook or courseware web pages.

Module 5805: Advanced Network Design

6 points

For a description see CSE5805 in the handbook or courseware web pages.

Module 5808: Quality of service in digital communication networks

6 points

For a description see CSE5808 in the handbook or courseware web pages.