
Note that these courses may be adjusted over the coming months which may involve some adjustments of the course content, learning objectives and summative assessment.

The list of courses is the currently planned one but it is subject to change.

CS2850 - Operating Systems

Prerequisites: CS1801 or CS1802

Course value: 15

COURSE SUMMARY

This course aims to introduce students to the principles of the function and architecture of operating systems, and also to give an understanding of how programs operate at system level. Course content includes:

Introductory topics: role of an operating system, computer architecture

Processes and threads: process management and scheduling, inter-process comunication, concurrency

Memory: partitioning, swapping and paging, caching, virtual memory, page replacement algorithms

File systems: implementation and maintenance

UNIX shell: starting programs, input and output streams, pipes, filters, utilities

System-level programming: memory handling, processes, threads, synchronisation, I/O

LEARNING OUTCOMES

By the end of this course a student should be able to:

- demonstrate an understanding of the principles of computer operating systems
- evaluate the theory and practice of existing operating systems
- demonstrate a working understanding of program execution, memory hierarchy, and the implementation of data structures
- understand system-level programming aspects such as memory management, interrupts, sockets and basic threading in C
- write simple shell scripts

TEACHING AND LEARNING METHODS

Lecture based delivery, supported by practical sessions. Normally 4 hours of lectures and laboratory classes per week.

KEY BIBLIOGRAPHY

- Andrew Tanenbaum, Herbert Bos: Modern Operating Systems, Pearson Education Limited, 4th edition, 2015.

- Randal E. Bryant and David R. O'Hallaron, Computer Systems: A Programmer's Perspective (3rd edition) Pearson, 2015, ISBN 978-0134092669

- W. R. Stevens, S. A. Rago: Advanced Programming in the UNIX Environment, Addison-Wesley Professional 2005 (2nd Edition)

- Brian W. Kernighan, Dennis M Ritchie: The C programming language, Englewood Cliffs, N. J., Prentice Hall 2nd ed. c1988

FORMATIVE ASSESSMENT AND FEEDBACK

Written grades will be given for the assessed assignments.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.

CS2855 - Databases

Prerequisites: CS1801 or CS1802

Course value: 15

COURSE SUMMARY

To provide the basic concepts of database technology.

To describe the need for database integrity and robustness.

To demonstrate the use of a modern database system in a web-based environment.

Course Content:

- Data modelling: views, subschema, data dictionary, data independence, entity relationship model.
- The relational model: relations, attributes, domains, relational algebra.
- Database design: normalisation, normal forms, entities and attributes
- SQL: basic SQL, correspondence between the relational model and SQL commands, simple queries, combination and sub-queries
- Administration and implementation: integrity, recovery from failure, concurrency, deletion and updating, forms, report writing. Programmatic access to SQL databases.

LEARNING OUTCOMES

- explain the issues involved in database design and the theory of the relational view of data
- describe the crucial issues concerning database integrity and recovery from failure
- write SQL queries
- be familiar with the steps for the design and implementation of a database, from the user specifications to the final design
- implement an interface to an SQL database using an API

TEACHING AND LEARNING METHODS

Lecture based delivery, supported by laboratory classes, guided independent study. Up to 4 hours of lectures and laboratory classes per week

KEY BIBLIOGRAPHY

- Database Systems Concepts (6th edition) A. Silbertschatz, H.F. Korth, S. Sudarshan McGraw Hill, 2011
- C.J. Date: An Introduction To Database Systems, 8th edition, Addison Wesley, 2003, ISBN 0321197844
- P. Rob and C. Coronel: Database Systems: Design, Implementation, And Management, Course Technology Inc, 2002, ISBN 061906269X
- R. Elmasri and S. Navathe: Fundamentals Of Database Systems, 3rd edition, Addison Wesley, 2000, ISBN 0805317554
- D.M. Kroenke: Database Processing, Prentice Hall, 1997, ISBN 0137378424

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback on progress will be given during the weekly laboratory sessions

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.

CS2860 - Algorithms and Complexity

Prerequisites: CS1801 or CS1802; CS1860

Course value: 15

COURSE SUMMARY

The main aim of the course is to teach the design of algorithms and data structures from the point of view of time and space complexity. This includes covering sorting and search algorithms, and graphs. The course content includes the following:

Complexity: counting, big-O notation, best-case, worst-case and average-case analysis.

Basic algorithms, sorting and searching part: implementation and analysis of linear search, binary search, and basic sorting algorithms, especially insertion sort, selection sort, merge sort, quick sort, heap sort.

Data structures: binary search trees, balanced binary search trees, hash tables, (binary) heaps.

Abstract datatypes: Sets, maps, priority queues.

Basic algorithms, graph algorithms part: adjacency matrix and adjacency list representations; algorithms for connectivity, shortest paths, and spanning trees.

LEARNING OUTCOMES

By the end of this course a student should be able to:

- understand and reason about alternative data structure representations, and their use in programs
- implement and reason about alternative implementations for basic algorithms, including graph algorithms
- calculate the complexity of basic algorithms, including graph algorithms

TEACHING AND LEARNING METHODS

Lecture based delivery.
Normally 3 hours of sessions per week.

KEY BIBLIOGRAPHY

- T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein: Introduction To Algorithms, 3rd Edition, MIT Press, Cambridge MA, 2009, ISBN 0262533058
- R. Sedgewick and K. Wayne: Algorithms, 4th edition, Addison-Wesley Professional

2011, ISBN 032157351X

- S. Skiena: The Algorithm Design Manual, 2nd edition, Springer 2010, ISBN 1849967202

- K. Mehlhorn and P. Sanders: Algorithms and Data Structures: The Basic Toolbox. Springer 2008.

  Available at http://www.mpi-inf.mpg.de/~mehlhorn/Toolbox.html

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback will be given during the laboratory sessions and feedback provided during lectures.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.

CS2900 - Multi-dimensional Data Processing

Prerequisites: CS1830; CS1860; cannot be taken with MT1820

Course value: 15

COURSE SUMMARY

During this course the student will be introduced to a progression of topics in Linear Algebra, starting from an introduction to vectors and matrices, moving to Singular Value Decomposition, the solution of linear equations and least squares, and then to eigenvalues and eigenvectors. The emphasis of this course will be computational, and a strong focus will be the Computer Science applications of the theoretical concepts covered.

LEARNING OUTCOMES

By the end of this course a student should be able to:

- Demonstrate an understanding of how to carry out vector operations such as dot product, length etc. on vectors in arbitrary dimensions. Demonstrate an understanding of the geometrical interpretation of vectors in 2 and 3 dimensions, and its applications in computer graphics.

- Demonstrate an understanding of linear transformations and their representation as matrices, of the application of matrices to vectors, and of matrix operations and their interpretation in 2, 3 and higher dimensions.

- Apply concepts such as matrix rank, transpose, upper and lower diagonal matrices, orthogonality, trace and symmetry.

- Demonstrate an understanding of Singular Value Decomposition, its numerical stability, and its relationship to the invertibility of a matrix.

- Demonstrate an understanding of the properties of eigenvalues and eigenvectors, and their construction for a given matrix.

- Throughout all the topics, demonstrate an understanding of the application of theoretical concepts and results in various areas of Computer Science.

TEACHING AND LEARNING METHODS

Lecture based delivery supported by laboratory sessions.
Normally 3 hours of sessions per week.

KEY BIBLIOGRAPHY

- Linear Algebra, A Modern Introduction, 4th Edition, David Poole, ISBN-13 978-1-285-46324-7

- Schaum's outlines Linear Algebra, Lipschutz and Lipson, McGraw Hill, ISBN-978-0-07-154353-8

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback will be given during the laboratory sessions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.

CS2910 - Artificial Intelligence

Prerequisites: None

Course value: 15

COURSE SUMMARY

Artificial Intelligence (AI) is usually defined as the science of making computers do things that require intelligence when done by humans. AI has had some success in limited, or simplified, domains. However, more recently, successes with developments of AI systems such as game playing and robotics have regenerated optimism concerning the attainment of human-level intelligence in a variety of domains despite the profound difficulty of the problem.

The aim of this course is to introduce students to the basic principles, methods and techniques of AI to provide the foundations for more advanced courses in this area. The course will start by providing an overview of the approaches in the field by referring to the wider historical context in which the AI vision was set and will further motivate the content to be taught by presenting existing and potential applications. After the overview, the course will deliver a series of topics from first principles, including the role of first-order logic for knowledge representation, computational reasoning and problem solving systems, the use of search as a capability for exploring alternative solutions, and how AI systems use knowledge to plan and learn from first principles.

Upon completion of the course, students should be able to develop intelligent systems by assembling capabilities to concrete computational problems; understand the role of knowledge representation, problem solving, and learning in intelligent-system engineering; and appreciate the role of problem solving, in wide applications that require a basic understanding of human intelligence from a computational perspective.

LEARNING OUTCOMES

By the end of this course a student should be able to:

- Use computational logic to model domains and reasoning tasks of an intelligent system
- Understand the role of knowldege representation, problem solving and learning in building domain independent and domain dependent AI capabilities
- Explain conceptual and computational trade-offs between the expressiveness of different representations and capabilities
- Demonstrate how to develop and combine AI capabilities in a suitable programming language

TEACHING AND LEARNING METHODS

Lecture based delivery supported by laboratory sessions.
Normally 2 hours of lectures per week for eleven weeks and 2 hours of labs for six weeks.

KEY BIBLIOGRAPHY

- Russell and Norvig. Artificial Intelligence: A Modern Approach. 3rd Edition.
- Ivan Bratko. Prolog Programming for Artificial Intelligence 3rd Edition.

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback will be given during the laboratory sessions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.

IY2760 - Introduction to Information Security

Prerequisites:

Course value: 15

COURSE SUMMARY

Course Content:

- Introduction: What is security (covering notions of Confidentiality, Integrity, and Availability)? Security threats and risks. Security management (ISO/IEC 17799). Data Protection legislation.

- Elements of cryptography: Ciphers (DES/AES). Message Authentication codes (MACs). Public key ciphers and digital signatures (RSA).

- Identity verification: use and storage of conventional passwords. Dynamic password schemes. Biometric techniques. Use of tokens (dumb and intelligent), including the use of secure elements such as smart cards and trusted execution environments (TEEs)

- Access control: Access Control Lists, capabilities, security labels (MAC and DAC), and role-based access control.

- CASE STUDY I: electronic payments (EMV). Examine the overall security functionality provided by widely utilised Europay-Mastercard-VISA (EMV) standard.

- Network security concepts: the concepts of security services and security mechanisms (as in ISO 7498-2) firewalls.

- Computer security: viruses, spyware, restricting access.

- Authentication and key distribution: The importance and relatedness of the concepts of key management and entity authentication in a network. Objectives of an entity authentication protocol. Some fundamental protocols (e.g. Kerberos). Using authentication protocols for key distribution, and other approaches to key establishment (including public key certificates and X.509).

- Cyber Physical security, Examine the security provisions, strengths and weaknesses of existing multi-application smart card platforms and operating systems along with the security of embedded systems and tokens.

- CASE STUDY II: Chip migration for financial institutions. Examine the relevant information security, business, design, architectural and other factors that may influence the adoption of chip card technology by financial institutions.

LEARNING OUTCOMES

On successful completion of this course, students will be able to:

- Identify, through the case studies how information security may be influenced by real world design and implementation decisions.
- Appreciate the different cryptographic algorithms, their use, advantages and disadvantages
- Apply the above identified cryptographic primitives in the review and evaluation of cryptographic protocols.
- Identify, through the case studies how information security may be influenced by real world design and implantation decisions.
- Appreciate the rational decisions in the design of a number tokens and secure elements

TEACHING AND LEARNING METHODS

33 hours of lectures. The coursework is designed to allow students to apply their knowledge to applications with which they are familiar.

KEY BIBLIOGRAPHY

- The main recommended text for this course is:
- D. Gollmann, Computer Security, John Wiley & Sons, 2005 (2nd edition).
- Useful background:
- C.P. Pfleeger and S. L. Pfleeger, Security in Computing, Prentice Hall, 2006 (4th edition).
- D. Saloman: Elements of Computer Security, Springer 2010
- K. Martin: Everyday Crytography, Oxford University Press 2012

FORMATIVE ASSESSMENT AND FEEDBACK

In class verbal feedback.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.

IY2840 - Computer and network security

Prerequisites: CS2850

Course value: 15

COURSE SUMMARY

This course is intended to provide a detailed exposition of computer and network security, which will build on knowledge acquired on previous computer science courses.

Course Content:

- Software vulnerabilities and hands on hacking-oriented attacks
- Memory errors
- Web
- Network
- Countermeasures
- Pointers to research papers

LEARNING OUTCOMES

- Identify and exploit the software vulnerabilities that can be introduced into programs through language features and poor programming practice
- Discuss the countermeasures that can mitigate the exploitation of such software vulnerabilities
- Introduce (briefly) malicious software (malware) as a typical consequence of a successful software exploitation
- Provide pointers to/discuss academic and /or industry research-oriented publications on the subject

TEACHING AND LEARNING METHODS

Lecture based delivery, guided independent study. Normally 3 hours of lectures per week.

KEY BIBLIOGRAPHY

- Hacking: The art of exploitation, 2nd edition, Jon Erickson

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback will be provided during laboratory sessions.

SUMMATIVE ASSESSMENT

Details of coursework submission deadlines will be published on the department website at the start of term.