
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.

CS3003 - IT Project Management

Prerequisites: CS2810 or CS2815

Course value: 15

COURSE SUMMARY

1. Projects and Project Work

2. Project Planning

3. Monitoring and Control

4. Change Control and Management

5. Quality

6. Estimating

7. Risk

8. Project Organisation

LEARNING OUTCOMES

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

- Apply basic project planning techniques
- Demonstrate an understanding of steps needed to build and maintain effective development teams
- Explain the procedures needed to monitor, control and report upon an IT development project
- Discuss and where appropriate apply the principles of project risk management
- Explain the ways in which appropriate quality attributes of the products of an IT development project can be assessed and assured

TEACHING AND LEARNING METHODS

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

KEY BIBLIOGRAPHY

- Field, Mike and Keller, Laurie, Project Management Open University, International

Thomson Press, 1998, ISBN: 1861522746

- Hughes, Bob and Cotterell, Mike, Software Project Management, McGraw-Hill (4th Ed), 2005, ISBN: 0077109899

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback during lectures. Written feedback from mandatory assessment

SUMMATIVE ASSESSMENT

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

CS3110 - Bioinformatics

Prerequisites: None

Course value: 15

COURSE SUMMARY

To introduce the main approaches currently in use in bioinformatics, with special emphasis on the analysis of DNA and protein sequences emerging from genome sequencing projects and genome-wide experimental assays.

Course Content:

- Basic molecular biology: introduction to the basic components of living cells, their functions and interactions, and to other concepts essential to understanding the use of computers in biology Sequence alignments, substitution matrices. Phylogenetic trees Dynamic Programming Systems biology: gene expression analysis, Protein-Protein Interaction analysis, biological networks, clustering

LEARNING OUTCOMES

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

- assess the main approaches currently in use in bioinformatics;
- demonstrate an understanding of the analysis of DNA and protein sequences.

TEACHING AND LEARNING METHODS

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

KEY BIBLIOGRAPHY

- Understanding Bioinformatics Marketa Zvalebil, Jeremy O. Baum Garland Science, 2007
- R.Durbin, S.Eddy, A.Krogh, and G.Mitchinson: Biological sequence analysis, Cambridge University Press, 1998 : ISBN-10: 0521629713
- R.Duda,P.Hart,D.Stork: Pattern classification, John Wiley & Sons, 2001:ISBN-10: 0471056693
- B.Levin: Genes VIII, Prentice Hall, 2004: ISBN-10: 0131239244
- A.M.Lesk: Introduction to Bioinformatics, Oxford University Press, 2005: ISBN-10: 0199277877

FORMATIVE ASSESSMENT AND FEEDBACK

Lectures in which students work through exercises getting verbal feedback.

SUMMATIVE ASSESSMENT

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

CS3220 - Digital Audio and Applications

Prerequisites: CS2800

Course value: 15

COURSE SUMMARY

Digital audio has many applications, ranging from music production to audio enhancement, or from noise cancellation in mobile phones to automatic music generations for games. These applications share common features that will be covered in this course. Starting with the fundamentals of sound (nature, transmission, volume), students will then learn how to convert sound to and from the digital domain. Once in the digital domain, audio can be manipulated using algorithms from Digital Signal Processing. The course covers some basic DSP algorithms, engaging the students both in audio analysis (amplitude, frequency, pan, etc.) and audio synthesis (additive , FM, etc.). By the end of this course, students should be able to write simple applications that analyse, manipulate and generate sound.

LEARNING OUTCOMES

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

- explain digital audio and the fundamentals of AD/DA conversion
- use Fourier analysis to synthesize and transform sounds in the frequency domain
- demonstrate an understanding lossy and losseless compression algorithms
- explain the basics of digital signal processing (post-processing and effects)
- develop applications of digital audio (music production, noise cancellation, audio enhancement, automatic music generation)
- acquire a basic knowledge of sound perception and psychoacoustics

TEACHING AND LEARNING METHODS

Up to 4 hours of lectures and laboratory classes per week.

KEY BIBLIOGRAPHY

- Musimathics, volumes 1 & 2, Gareth Loy, The MIT press, 2011

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback during laboratory sessions.

SUMMATIVE ASSESSMENT

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

CS3250 - Visualisation and Exploratory Analysis

Prerequisites:

Course value: 15

COURSE SUMMARY

The course aims to teach the principles and arts of statistical visualisation and exploratory analysis of data.

Course Content:

- Construction of informative bivariate plots.
- Visualisation of multivariate data
- Dimensional reduction. Non-linear methods (t-SME, isomap, Proxigrams)
- Exploratory cluster analysis
- Standard methods for visualisation of relational and graph data (Gephi)
- Importance of guarding against "snooping"
- Basic principles of colour scale design and glyph choice

LEARNING OUTCOMES

By the end of the course students should be able to:

- Have some of the skills and wisdom that are needed for open-ended exploratory analysis of data, and for the analytical presentation of the results of statistical analyses
- Understand and be able to construct linear projections of multivariate data; they should also have experience in using some non-linear dimension reduction methods
- have some practical experience of using standard graph visualisation methods and evaluation of results
- Be aware of the dangers of data snooping, through practical experience of this
- Be able to make reasoned choices in representational mode, glyph design, and colour design for presentation graphics

TEACHING AND LEARNING METHODS

Lecture based delivery, laboratory classes, guided independent studyNormally 3 hours of lectures / laboratory per week

KEY BIBLIOGRAPHY

- Visualising data, by William S. Cleveland
- The elements of graphing data by William s. Cleveland
- The visual display of quantitative data, by Edward Tufle
- Envisioning information by Edward Tufle

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback during laboratory sessions.

SUMMATIVE ASSESSMENT

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

CS3470 - Compilers and code generation

Prerequisites: CS1801 or CS1802, CS1870

Course value: 15

COURSE SUMMARY

To describe how to construct and implement interpreters and compilers for modern processors

Course Content:

- Lexical analysis: input buffering, regular expressions, finite state automata, hash coded symbol tables, LEX
- Syntax analysis: context free grammars, derivations, recursive descent and table-based LR parsing, YACC Semantics: abstract parse trees, syntax directed translation, intermediate forms and three address code, attributes, annotated parse trees, semantic rules and attribute grammars, translation schemes and L-attribute grammars
- Error detection and recovery: classes of error and error recovery in top down parsers
- Intermediate code generation: using attribute grammars, three address code
- Code improvement: basic blocks and code improvement techniques, flow graphs, loop improvement and fusion, directed a cyclic graphs for identifying code improvements
- Complier generation with tool support

LEARNING OUTCOMES

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

- explain the role and structure of a compiler and the standard stages of compilation
- to build a DFA based lexical analyser for a set of tokens specified using regular expressions
- Advanced students should be able to construct regular expressions, which define specified set of strings
- use grammars to define context free languages and to build parsers for them
- describe syntax directed translators and use them to construct intermediate code

- describe various types of error detection and recovery
- generate three address code from source code.

TEACHING AND LEARNING METHODS

Lecture based delivery, supported by small group tutorials.Normally 3 hours of lectures per week

KEY BIBLIOGRAPHY

- Aho, Lam, Sethi, Ullman: Compilers: Principles, Techniques, and Tools, Addison Wesley, 2006 ISBN: 10: 0321486811
- J. Tremblay and P.G. Sorenson: The theory and practice of compiler writing, McGraw Hill, 1985, ISBN 0-07-065161-2
- N. Wirth: Algorithms + Data Structures = Programs, Prentice Hall, 1976.

FORMATIVE ASSESSMENT AND FEEDBACK

Return of coursework grades and comments, in-class discussion of coursework solutions

SUMMATIVE ASSESSMENT

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

CS3480 - Software Language Engineering

Prerequisites: CS1801 or CS1802; CS2810 or CS2815

Course value: 15

COURSE SUMMARY

Course Content:

- Domains
- Domain Specific Languages
- Concrete and abstract syntax, generalised vs near deterministic parsing
- Code generation
- The ART toolset
- Case studies

LEARNING OUTCOMES

- understand domain specific language design and implementation
- learn the significance of DSL's and meta-modelling in advanced software engineering
- design and implement small specialised languages

TEACHING AND LEARNING METHODS

Up to 3 hours of lectures and laboratory sessions per week.

KEY BIBLIOGRAPHY

- Software Language Engineering: Creating Domain-Specific Languages Using Metamodels, Anneke Kleppe,Addison-Wesley Professional, 2008, ISBN-10:0-321-55345-4
- Metamodelling for software engineering Cesar Gonzalex-Perez, Brian Henderson-Sellers, wiley Agust 2008, ISBN 978-0-470-03036-3

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.

CS3490 - Computational Optimisation

Prerequisites: CS2860

Course value: 15

COURSE SUMMARY

To introduce the basic models of computational optimisation and the basic algorithms for solving computational optimisation problems. To demonstrate the theoretical and computational methods of analysing computational optimisation algorithms and will discuss available software packages for solving problems.

Course Content:

- Introduction: algorithm efficiency and problem complexity
- Linear programming (LP): LP model, formulating problems as LP problems, graphical solution, simplex method, duality in LP;, decomposition of LP problems, LP software
- Integer Programming (IP): IP models, branch-and-bound algorithm
- Computational optimisation problems: greedy-type algorithms, construction heuristics and local search for the TSP
- Heuristics: DMERN problem and signed graphs; heuristics for DMERN, experimental analysis of embedded network and TSP heuristics, theoretical analysis of heuristics, meta-heuristics

LEARNING OUTCOMES

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

- demonstrate an understanding of the basic models of computational optimisation
- apply the basic algorithms for solving computational optimisation problems
- evaluate theoretical and computational methods of analysing computational optimisation algorithms
- use enhanced algorithmic and mathematical skills.

TEACHING AND LEARNING METHODS

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

KEY BIBLIOGRAPHY

- M.W. Carter and C.C. Price: Operations Research: A practical Introduction, CRC, 2001 (001.424 CAR): ISBN-10: 0849322561

- Z. Michalewicz and D.B. Fogel: How to Solve It: Modern Heuristics, Springer, 2000: ISBN-10: 3540224947

- F. Glover and M. Laguna: Tabu Search, Kluwer, 1997: ISBN-10: 079239965X

- J. Bang-Jensen and G. Gutin: Digraphs: Theory, Algorithms and Applications, Springer, 2000: ISBN-10: 1852336110

FORMATIVE ASSESSMENT AND FEEDBACK

Return of marked zero-weighted assignments. Some solutions discussed in class.

SUMMATIVE ASSESSMENT

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

CS3510 - Functional Programming and Applications

Prerequisites:

Course value: 15

COURSE SUMMARY

This course introduces the principles of functional programming (FP). Functional programming has become more and more popular in recent years because it promotes code that's safe, concise, and elegant, and makes it easier to write parallel code for parallel computing. You will also study how FP may be combined with other programming paradigms (eg, OOP) in languages such as Scala, a language that fuses FP and OOP in a practical package for applications.

Course Content:

- Introduction to the basic principles of functional programming (FP) as compared to the imperative object-oriented programming such as Java
- Some basic features in an FP language: recursion, abstraction and higher-order functions
- Basic typing and data types such as lists and tuples
- High-order functions and pattern matching
- Classes and objects in a hybrid language (such as Scala)
- Functional paradigm as compared with the imperative paradigm
- Applications (eg. In parallel programming)
- Basics in some advanced topics such as genericity, monads and dependent typing

LEARNING OUTCOMES

TEACHING AND LEARNING METHODS

Normally 3 hrs per week of lectures and practicals.

KEY BIBLIOGRAPHY

- M. Odersky, L. Spoon and B. Venners. Programming in Scala. 2nd ed. Dec 2010.
- S. Thompson. Haskell: The Craft of Functional Programming. 3rd ed. 2012.

- C.S. Horstmann. Scala for the Impatient. Addison-Wesley 2012

FORMATIVE ASSESSMENT AND FEEDBACK

Written assignment with written feedback

SUMMATIVE ASSESSMENT

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

CS3846 - Human-Computer Interaction

Prerequisites: CS1840 or CS2841

Course value: 15

COURSE SUMMARY

This course introduces human-computer interaction aspects and challenges, and addresses the approaches that can be used to create interfaces matching users' needs and expectations.

Course Content:

- introduction to Human-Computer Interaction (definition and history);
- user experience (UX) vs. user interface (UI);
- heuristic evaluation;
- rapid prototyping;
- HCI studies/experiments.

LEARNING OUTCOMES

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

- demonstrate an understanding of what HCI is, its principles, challenges, and teamwork development aspects
- build prototypes using different media and technologies
- evaluate the usability and suitability of user interfaces
- undertake HCI tests, namely Web experiments
- explain how perception and cognition influence HCI

TEACHING AND LEARNING METHODS

Up to 4 hours of lectures and laboratory classes per week

KEY BIBLIOGRAPHY

- Alan Dix et al, Human Computer Interaction. 2003.
- Yvonne Rogers et al, Interaction Design: Beyond Human Computer Interaction. 2011.

- Jakob Nielsen, Prioritizing Web usability. 2006.
- Don Norman, The Design of Everyday Things: Revised and Expanded Edition. 2013.
- Eric Freeman. Head First HTML5 Programming. 2011.
- Jon Duckett, JavaScript and jQuery: Interactive Front-End Web Development. 2014.

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.

CS3920 - Machine Learning

Prerequisites: A-level Mathematics

Course value: 15

COURSE SUMMARY

The course will teach main ideas of machine learning with a particular emphasis on kernel methods.

Course Content:

- Nearest neighbours for classification and regression: interesting distances
- Discriminant analysis
- Ridge regression and Lasso
- Support vector machines for classification and regression
- Kernel trick and its applications to the algorithms covered so far
- Practical useful kernels, including string kernels.

LEARNING OUTCOMES

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

- demonstrate knowledge of the theoretical background in machine learning methods
- have a basic understanding of the main advantages and limitations of various approaches to machine learning and specific machine-learning algorithms
- be able to implement basic versions of several machine-learning algorithms
- have a basic understanding of some ways to apply the ideas and algorithms of machine learning in industry, medicine and other fields

TEACHING AND LEARNING METHODS

Lecture based delivery, supported by laboratory classes.Normally 3-4 hours of lectures/labs per week. Guided independent study

KEY BIBLIOGRAPHY

- Gareth James et al. "An introduction to Statistical learning with R" Springer, 2013

- Trevor Hastie et al. "The elements of Statistical learning", Springer, 2009
- John Shawe-Taylor and Nello Cristianini: Kernel Methods for Pattern Analysis, Cambridge University Press, 2004: ISBN-10: 0521813972

FORMATIVE ASSESSMENT AND FEEDBACK

In lectures / labs, Immediate help and verbal feedback

SUMMATIVE ASSESSMENT

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

CS3930 - Computational Finance

Prerequisites: A-level Mathematics

Course value: 15

COURSE SUMMARY

To understand the key role played by the advent of derivatives, financial instruments which facilitate managing financial risks. To understand pricing derivatives (and associated strategies of dynamic hedging) using advanced computational models are required.

Course Content:

- Introduction: financial markets; the rules of the game.
- Taxonomy of securities: main kinds of derivative securities and underlying markets.
- Mathematical techniques: Wiener process; diffusion processes as mathematical models of price dynamics; stochastic differential equations; computer simulations.
- Pricing and hedging in the Black-Scholes world: risk-neutral valuation; the Black-Scholes equation and analytic formulae; the "Greeks" and their use.
- Beyond the Black-Scholes world: application issues; computational models; fractals and their use in finance.
- Efficient markets hypothesis: theory vs empirical evidence.
- Risk management: Value at Risk.
- Coursework Project: implementing valuation algorithms for different derivatives (e.g., in MatLab); solving numerical and theoretical problems.

LEARNING OUTCOMES

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

- demonstrate an understanding of mathematical and computational models of underlying and derivative securities;
- master techniques for pricing derivatives and for dynamic hedging;
- apply these models and techniques for creating computer programs.

TEACHING AND LEARNING METHODS

Lecture based delivery, supported by practical classes.Normally 3 hours of lectures and

laboratory classes per week.

KEY BIBLIOGRAPHY

- John Hull, Options, Futures and Other Derivatives, 6th edition, Prentice-Hall, 2007.
- ISBN-10: 0131977059 (earlier editions starting from the 2nd are acceptable)
- Paul Wilmott Introduces Quantitative Finance, John Wiley, 2001. ISBN-10: 0471498629
- Paul Wilmott, Sam Howison, and Jeff Dewynne, The Mathematics of Financial Derivatives: A Student Introduction, Cambridge University Press, 1995. ISBN-10: 0521497892

FORMATIVE ASSESSMENT AND FEEDBACK

Return of grades and written comments on courseof grades and written comments on courswork projecte submission of courseworkwork project

SUMMATIVE ASSESSMENT

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

CS3940 - Intelligent agents and multi-agent systems

Prerequisites: None

Course value: 15

COURSE SUMMARY

The aims of this course include introducing the students to the concept and design of an agent and multi-agent system, and the main applications for which they are appropriate. In addition, it presents a contemporary platform for implementing agents and multi-agent systems. Course contents include:

Introduction: agents and objects, expert systems, distributed systems; typical application areas for agent systems.

Intelligent Agents: abstract architectures for agents; tasks for agents, the design of intelligent agents - reasoning agents, agents as reactive systems; hybrid agents (e.g., PRS); layered agents (e.g., Interrap).

Multi-Agent Systems: classifying multi-agent interactions - cooperative versus non-cooperative; zero-sum and other interactions; cooperation - the Prisoner's dilemma and Axelrod's experiments; interactions between self-interested agents: auctions systems; negotiation; argumentation; interaction languages and protocols: speech acts, KQML/KIF, the FIPA framework, ontologies, coordination languages; interactions between benevolent agents: cooperative distributed problem solving (CDPS), partial global planning; coherence and coordination; applications of intelligent agents and multi-agent systems.

LEARNING OUTCOMES

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

- understand the notion of an agent and understand the characteristics of applications that lend themselves to an agent-oriented solution
- understand the key issues associated with constructing agents capable of intelligent autonomous action
- understand the key issues in designing societies of agents that can effectively cooperate in order to solve problems
- understand the main application areas of agent-based solutions, and be able to develop a meaningful agent-based system

TEACHING AND LEARNING METHODS

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

KEY BIBLIOGRAPHY

- M. Wooldridge: An Introduction to MultiAgent Systems. John Wiley & Sons (2nd edition) 2009: ISBN 978-0-470-51946-2

- G. Weiss, editor: Multi-Agent Systems, A Modern Approach to Distributed Artificial Intelligence. The MIT Press, 1999: ISBN-10: 0262232030

- J. Ferber: Multi-Agent Systems. Addison-Wesley, 1999: ISBN-10: 0201360489

- M. Singh and M. Huhns: Readings in Agents. Morgan-Kaufmann Publishers, 1997: ISBN-10: 1558604952

FORMATIVE ASSESSMENT AND FEEDBACK

Verbal feedback provided through the tutorial sessions.

SUMMATIVE ASSESSMENT

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

CS3945 - Semantic Web

Prerequisites: CS1840 or CS2841

Course value: 15

COURSE SUMMARY

The Web, as it exists today, primarily supports human understanding and the interpretation of the vast information space it encompasses. However the Web was originally designed with a goal to support not only human-human communication but also as one that would enable automated machine processing of data with minimal human intervention. The Semantic Web is Tim Berners-Lee's vision of a machine understandable and unambiguously computer interpretable Web. The rationale behind such a system is that most of the data currently posted on the web is buried in HTML files suitable for human reading and not for computers to manipulate meaningfully. The semantic Web, an extension of the current web, can be thought of as a globally linked database where information is given well-defined meaning using metadata for better enabling computers and humans to work in close cooperation. The realisation of a Semantic Web will thus make machine reasoning more ubiquitous and powerful, creating an environment where intelligent software agents can roam, carrying out sophisticated tasks for their users.

Though the original motivation of the semantic web was to constitute the next generation of the WWW, the standards and technologies developed in the process have been found useful in specific realm enterprises as well. From this perspective the Semantic Web can be viewed as a semantically-rich data model that is more expressive than the usual relational data model used in standard databases systems, and is also more adequate to distributed and incomplete resources.

This course is about the notions, concepts, technologies and modelling techniques that constitute the Semantic Web, whose key distinguishing characteristics will be the support for and use of semantics in new, more effective, more intelligent, ways of managing information and supporting applications.

LEARNING OUTCOMES

- explain fundamental concepts, and standards of the semantic web
- demonstrate a deep understanding of the use of standards such as RDF, RDFS and OWL for modeling different scenarios and reasoning
- demonstrate a deep understanding of underlying logical theory behind the semantic web, for example Description Logic

TEACHING AND LEARNING METHODS

The course will be presented as two hours of lectures per week, and one hour per week of practical classes during five weeks.

KEY BIBLIOGRAPHY

Core reading material:

- Semantic Web for the Working Ontologist (Effective Modeling in RDFS and OWL) 2nd Edition, By Dean Allemang, James Hendler, Elsevier / Morgan Kaufmann, July 2011.

- A Semantic Web Primer (Cooperative Information Systems), 3rd Ed., by Grigoris Antoniou (Author), Paul Groth (Author) et al., 2012.

- Semantic Web Programming, John Hebeler, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez, Wiley Publishing, Inc, 2009

FORMATIVE ASSESSMENT AND FEEDBACK

The students will receive oral feedback in laboratory sessions.

SUMMATIVE ASSESSMENT

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

IY3606 - Smart Cards, RFIDs and Embedded Systems Security

Prerequisites: IY2760 or equivalent

Course value: 15

COURSE SUMMARY

Students will have developed a mastery of the topic of smart cards and token security.
Course contents:
Introduction to Smart Cards/Chips & RFID/NFC; Assets for Cyber Security
Smart Cards – Trusted Production Environment
Operating systems, Interoperability and Security
Applications & Security for Mobile Communications, USIM/SIM, Services and Clouds
Smart Cards for Secure Banking & Finance
Smart Cards in eIDs/Passports - & RFIDs/NFC explained
Advances in Smart Chips/Tokens, and Transport System Case Study
Common Criteria and Smart Cards
Security Attacks, Countermeasures and Testing for Smart Cards/RFIDs/NFC
Application Development Environments for JAVA and SIM Toolkit
Comparing Alternative Security Tokens/Environments; including TPM and TEE

LEARNING OUTCOMES

- identify constituent components, analyse strengths and weaknesses, identify new applications of smart cards/security tokens and their use as assets in cyber security; identify the steps in the manufacturing/personalisation processes, analyse and evaluate potential risks and compare security safeguards

- identify and compare the systems in use, analyse the strengths and weaknesses and evaluate interoperability and security issues; analyse the range of capabilities of SIM/USIM cards in Smartphones and apply them to new service ideas, evaluate the possible range of services and security measures

- understand the main standards and applications of smart cards for banking and finance, compare with earlier card solutions and analyse strengths and weaknesses of approaches; analyse the key role of the smart card/RFID for passports, IDs and satellite TV, evaluate the security measures that have protected past and current cards,

- identify and describe new technologies, including NFC, TPM, TEE; and apply them to new applications and evaluate the likely suitability/success of approach; explain how common criteria may affect smart card design/development, analyse the different approaches and compare with less formal methods

- identify and describe the classes of attack and notable methods within each class, analyse countermeasures and evaluate practicality of attacks and the

effects on cyber security; identify, compare and evaluate different methods of developing applications for smart cards, and understand the development cycle and the use of practical tools

- analyse the issues concerning smart card lifecycle management, and evaluate and compare methods of local and remote card management

TEACHING AND LEARNING METHODS

Lecture based delivery. Normally 3 hour of lectures per week.

KEY BIBLIOGRAPHY

- Keith Mayes, Konstantinos Markantonakis, "Smart Cards, Tokens, Security and
- Applications", Springer-Verlag New York, January 2008, ISBN: 0387721975
- W. Rankl and W. Effing – "Smart card handbook" 2nd edition John Wiley 1997
- Klaus Finkenzeller, "The RFID Handbook", John Wiley and Sons 2003
- Zhiqun Chen, "Java Card Technology for Smart Cards", Addison- Wesley 2000.
- Keith Mayes, Konstantinos Markantonakis, "Secure Smart Embedded Devices, Platforms and Applications", Springer-Verlag New York, 2013, ISBN 978-1-4614-7914-7

FORMATIVE ASSESSMENT AND FEEDBACK

Oral feedback during lectures.

SUMMATIVE ASSESSMENT

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

IY3609 - Digital Forensics

Prerequisites: IY2760 or equivalent

Course value: 15

COURSE SUMMARY

Introduction to forensic science, steps from collecting data to preserving evidence, and a framework for digital forensic evidence collection and processing.
Fundamentals of host forensics for Microsoft Windows, including kernel architecture, device driver architecture, registry, auditing, and security architecture, file system handling, and reconstruction of file and directory structures on the FAT and NTFS file system families.
Fundamentals of host forensics for Unix derivatives using the Linux operating system as an exemplar, including kernel and device driver architecture, security and audit mechanisms, file systems and pseudo file systems, and the reconstruction of file and directory structures using UFS and Ext2/3fs as exemplars.
Foundations of network forensics from data capturing and collection to network file systems and supplementary protocols as well as selected application-layer protocols and techniques used for identifying and reverse-engineering protocols used on networks.
Introduction to malware including anti-forensics and propagation techniques.
Introduction to steganographic techniques for images, video, textual data, and audio as well as steganalytical techniques for selected media types and approaches to traitor tracing.
A survey of non-standard storage mechanisms from retention characteristics to mobile and smart phones and vehicular systems as well as network-based search and storage mechanisms.

LEARNING OUTCOMES

- understand the concepts of audit and indirect activity records retained by operating systems, particularly in file systems, and how to retrieve such information
- understand selected network protocols and the collection and derivation of evidence leading to the reconstruction of system and user activity based on network trace information
- understand infiltration and anti-forensics techniques used particularly by malicious software
- apply steganographic and particularly steganalytical methods for different types of media
- understand the retention characteristics of storage systems and non-standard devices such as mobile/smart phones, cloud computing, and vehicular systems

TEACHING AND LEARNING METHODS

Lecture based delivery. Normally 3 hour of lectures per week.

KEY BIBLIOGRAPHY

- K.J. Jones, R. Bejtlich, C. W. Rose: Real Digital Forensics. Addison-Wesley, 2006
- B. Carrier: File System Forensic Analysis. Addison-Wesley, 2005
- D. P. Bovet, M. Cesati: Understanding the Linux Kernel, 3rd ed. O'Reilly, 2006
- M. Russinovich, D.A. Soiomon, A. Ionescu: Windows Internals, 5th ed. Microsoft Press, 2008

FORMATIVE ASSESSMENT AND FEEDBACK

Oral feedback during lectures.

SUMMATIVE ASSESSMENT

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

IY3612 - Cyber Security

Prerequisites: IY2760 or equivalent

Course value: 15

COURSE SUMMARY

Week 1: Introduction
Week 2: Networks and dependencies
Week 3: Critical infrastructures and interdependencies
Week 4: Security of Cyber-Physical Systems
Week 5: Control systems security
Week 6: Advanced persistent threats
Week 7: Attack modelling techniques
Week 8: System assurance
Week 9: Incident response mechanisms
Week 10: Offensive cyber operations
Week 11: Future challenges

LEARNING OUTCOMES

- have an understanding of network robustness and failures, together with key underlying theoretical concepts
- understand critical (information) infrastructures, vulnerabilities, and their dependencies
- appreciate the specific security problems of cyber-physical including SCADA systems and selected infrastructure
- understand complex attacks, analytical models for such attacks, and assurance mechanisms

TEACHING AND LEARNING METHODS

Lecture based delivery. Normally 3 hour of lectures per week.

KEY BIBLIOGRAPHY

- J Lopez, R Setola, S Wolthusen (edsCritical Information Infrastructures Vol. 7130 of Lecture Notes in Computer Science. Springer-Verlag, 2012

FORMATIVE ASSESSMENT AND FEEDBACK

Oral feedback during lectures.

SUMMATIVE ASSESSMENT

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

IY3660 - Applications of Cryptography

Prerequisites: IY2760

Course value: 15

COURSE SUMMARY

The aim of this module is to explain the fundamentals behind cryptography and how it is deployed in real systems.

Course Content:

- Fundamentals: security services, security models, basic attacks on cryptosystems
- Cryptographic mechanisms: symmetric and public-key encryption, hash functions, MACs, digital signatures, authentication protocols
- Real world cryptography: key management, implementation issues, cryptographic standards, crypto politics
- Applications: case studies of systems such as 3G, EMV, SSL/TLS

LEARNING OUTCOMES

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

- Appreciate the full range of security services that can be provided by cryptography
- Compare a number of different cryptographic mechanisms that can be used to provide these security services
- Explain the full nature of the architecture within which cryptography is deployed in a security system
- Identify appropriate cryptographic standards that should be followed when implementing cryptography
- Explain the rationale for the design decisions taken in a number of widely deployed cryptographic systems

TEACHING AND LEARNING METHODS

Up to 3-4 hours of lectures and laboratory sessions per week.

KEY BIBLIOGRAPHY

- D.R. Stinson, Cryptography: Theory and Practice, CRC Press

- K.M. Martin: Everyday Cryptography, OUP.
- A. Dent and C.J. Mitchell: A users' Guide to Cryptography Standards, Artech House
- C. Paar and J. Pelzl, Understanding Cryptography, Springer
- J. Katz and Y. Lindell, Introduction to Modern Cryptography, CRC Press
- N. Ferguson, B. Schneier and T. Kohno, Cryptography Engineering, Wiley

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.

IY3840 - Malicious Software

Prerequisites: IY2760 and IY2840

Course value: 15

COURSE SUMMARY

Cybercrime has become both more widespread and harder to battle. Researchers and anecdotal experience show that the cybercrime scene is becoming increasingly organized and consolidated, with strong links also to traditional criminal networks. Modern attacks are indeed stealthy and often profit oriented.
Malicious software (malware) is the traditional way in which cybercriminals infect user and enterprise hosts to gain access to their private, financial, and intellectual property data. Once stolen, such information can enable more sophisticated attacks, generate illegal revenue, and allow for cyber-espionage.
By mixing a practical, hands-on approach with the theory and techniques behind the scene, the course discusses the current academic and underground research in the field, trying to answer the foremost question about malware and underground economy, namely, "Should we care?".
Students will learn how traditional and mobile malware work, how they are analyzed and detected, peering through the underground ecosystem that drives this profitable but illegal business. Understanding how malware operates is of paramount importance to form knowledgeable experts, teachers, researchers, and practitioners able to fight back. Besides, it allows us to gather intimate knowledge of the systems and the threats, which is a necessary step to successfully devise novel, effective, and practical mitigation techniques.

LEARNING OUTCOMES

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

- Comprehend what malicious software is
- Understand the malware landscape of the early days and what efforts and challenges the AV industry was facing to fight malware threats; (basic) static analysis to analyze and detect malware; (basic) assembly and reverse engineering notions are provided
- Explain dynamic analysis. Packing and algorithmic-agnostic unpacking is introduced as an initial step toward full dynamic analysis; sandboxes and limits of dynamic analysis and sandboxes.
- Describe and explain the underground economy and cybercrime that surrounds malicious software
- Demonstrate a critical appreciation of some of the newer research trends that are likely to influence software security work in the coming years

TEACHING AND LEARNING METHODS

Up to 3-4 hours of lectures and laboratory sessions per week.

KEY BIBLIOGRAPHY

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.