Eötvös Loránd University
Faculty of Informatics

Computer Science Msc

1. Structure of the course

To fulfil the course requirements of the Computer Science MSc course, you have to:

- pass all compulsory subjects listed in the curriculum.
- study and pass some further courses (so-called elective courses) in the amount of 7 credits.

The courses are organised in 4 semesters. The courses of the 2nd and 4th semesters are typically offered only in the Spring semesters, whereas the courses of the 1st and 3rd semsteres are offered only in the Autumn semesters. Students starting their studies in Spring semester in February are instructed to fulfil the semesters of the Computer Science MSc programme in the following order: 2nd, 1st, 4th, 3rd. That is to say, they are obliged to fulfil firstly the subjects of the second semester, and so on.

The elective courses that belong to the Computer Science MSc programme are listed below. As a Computer Science MSc student at ELTE, one can choose from further elective courses and study various subjects and get credits for passing them: courses of our Computer Science BSc programme, cartography/geoinformatics courses, language courses, sport courses.

1st semester	**2nd semester**	**3rd semester**	**4th semester**
Obligatory courses			
Data mining and information retrieval (5 credits)	Complex information systems (5 credits)	Advanced Java programming (5 credits)	Thesis Work (30 credits)
Interactive media design and development (5 credits)	Formal semantics (3 credits)	Analysis of distributed systems (5 credits)	Software engineering lab 2. (5 credits)
Models of computation (5 credits)	Functional languages (5 credits)	Design of distributed systems (5 credits)	
Preparation course for master studies and developing learning skills (0 credits)	Service science (5 credits)	Scalable enterprise applications (5 credits)	
Theory of programming (5 credits)	Software quality and testing (5 credits)	Software engineering lab 1. (5 credits)	
Web engineering (5 credits)	Software technology (5 credits)		
Elective courses			
elective course (5 credits)	elective course (2 credits)		
total credits			
30 credits	**30 credits**	**25 credits**	**35 credits**

2. List of courses with their credit values and weekly hours

„L." stands for lecture and „Pr." denotes practice.

1 contact hour = 45 minutes

RS: semester, in which the given course is recommended
SO: semester, in which the given course is typically offered
LH/W: lecture contact hours per week
PH/W: practice contact hours per week

Title of the course	Credits	RS	SO	LH/W	PH/W
Advanced Java Programming L.	2	3	Autumn	2	
Advanced Java Programming Pr.	3	3	Autumn		2
Analysis of distributed systems L+Pr.	5	3	Autumn	2	2
Complex information systems L.	2	2	Spring	2	
Complex information systems Pr.	3	2	Spring		2
Data mining and information retrieval L.	2	1	Autumn	2	
Data mining and information retrieval Pr.	3	1	Autumn		2
Design of distributed systems L.	2	3	Autumn	2	
Design of distributed systems Pr.	3	3	Autumn		2
Formal semantics	3	2	Spring	2	
Functional languages L+Pr.	5	2	Spring	2	2
Interactive media design and development L+Pr.	5	1	Autumn	2	2
Models of computation L.	2	1	Autumn	2	
Models of computation Pr.	3	1	Autumn		2
Preparation course for master studies and developing learning skills	0	1	Autumn, Spring		2
Scalable enterprise applications L.	2	3	Autumn	2	
Scalable enterprise applications Pr.	3	3	Autumn		2
Service Science L.	2	2	Spring	2	
Service Science Pr.	3	2	Spring		2
Software engineering lab 1.	5	3	Autumn, Spring		3
Software engineering lab 2.	5	3	Autumn, Spring		3
Software quality and testing L.	2	2	Spring	2	
Software quality and testing Pr.	3	2	Spring		2
Software Technology L+Pr.	5	2	Spring	2	2
Theory of programming L.	2	1	Autumn	2	
Theory of programming Pr.	3	1	Autumn		2
Web engineering L.	2	1	Autumn	2	
Web engineering Pr.	3	1	Autumn		2

Title of the course	Credits	RS	SO	LH/W	PH/W
Advanced cryptography L.	2		Autumn	2	
Advanced cryptography Pr.	4		Autumn		2
Applied cryptography project seminar L.	2		Autumn	2	
Applied cryptography project seminar Pr.	4		Autumn		2
Cryptographic protocols L.	2		Autumn	2	
Cryptographic protocols Pr.	2		Autumn		2
Economics of security and privacy L.	4		Autumn	2	
Learning technologies Pr.	3		Autumn, Spring		2
Programming in Erlang	2		Spring		2
Network Algorithms	5		Autumn	2	2

3. Description of the courses

Advanced Java programming
A short description of the course, topics: The purpose of the course is to acquire knowledge on, and enhance competence in, Java Standard Edition, beyond the fundamental language concepts and standard libraries: Generic definitions Annotations Reflection Multithreading Memory management, garbage collection Input-output, serialization Database management and persistence: JDBC and the fundamentals of JPA Network programming: TCP and UDP; HTTP Program design principles and best practices Exceptions, assertions Logging and testing
Literature: • James Gosling, Bill Joy, Guy Steele, Gilad Bracha. The Java Language Specification, Third Edition. Addison-Wesley, 2005. ISBN 0-321-24678-0 • Linda DeMichiel, Michael Keith. JSR 220: Enterprise JavaBeans, Version 3.0, Java Persistence API. Sun Microsystems, Inc., 2006.

Analysis of distributed systems
A short description of the course, topics: The goal of the subject is to give an overview for the student about how can we explain the parallel behaviour by algebraic methods and Petri-nets, and how work applications based on that models in practice. The basic concepts of the course are processes, computational processes, parallelism, operations of processes, compositions of processes and properties of processes (liveness, deadlock-free, etc.). The theory of Petri-nets is explored more partially with many modelling example. The behavioural and structural properties, methods of analysis, famed subclasses and relationships between these subclasses are investigated. We define theorems about liveness, safetyness and reachability and present transformation, which preserve these properties. The course introduces the Petri-boxes, a special class of Petri-nets, which help us to model the program structures (sequences, branches and loops). Some tools for simulation and analysis of Petri-nets are also investigated. The second part of the course introduces the theory of algebraic models through a given example. The properties of the models, the methods of descriptions of processes and the possible compositions are examined. The denotational, operational and axiomatic semantics of the model is given and the relationships of these different descriptions are investigated. Teaching methods: There will be lectures introducing the formal specification and properties of Petri nets and algebraic models and exercises where the students will create concrete examples. There will be also programming exercises where the students can use the learned methods.
Literature: • Murata, T.: Petri Nets, Properties, Analysis and Applications (Proc. of the IEEE. Vol. 77., no. 4, ASpr 1989, 541-580) • Best, E., Devillers, R., Koutny, M.: Petri Net Algebra (Springer 2001) • Hennessy M.: Algebraic Theory of Processes (MIT, 1989) • Hoare, C.A.R.: Communicating Sequential Processes (Prentice-Hall, 1985)

Complex information systems
A short description of the course, topics: Concept of Information Systems Methodologies for Analysing and Designing Information Systems Concept of Enterprise Resource Systems Logistics as Business Process of Enterprises

Human Resources Management
WFMS (Workflow Management Systems)
Integration problem. Different approaches. Current trends.
Cloud Computing

Literature:

- Shelly, G. B., & Rosenblatt, H. J. (2012). „*System Analysis and Design 9th Edition* ". *Shelly Cashman Series*.
- Scheer 1994, August-Wilhelm Scheer, *Business Process Engineering* Study Edition: Reference Models for Industrial Enterprises, Springer-Verlag ,1994
- Magal, S. R., & Word, J. (2011). *Integrated business processes with ERP systems*. Wiley Publishing.
- Magal, S. R., & Word, J. (2009). *Essentials of business processes and information systems*. Wiley Publishing.
- Curran, T., Keller, G., & Ladd, A. (1997). *SAP R/3 business blueprint: understanding the business process reference model*. Prentice-Hall, Inc.
- Langer, A. M. (2007). *Analysis and design of information systems*. Springer Science & Business Media.
- Larman, C. (2012). *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Interative Development*. Pearson Education India.
- Bennett, S., McRobb, S., & Farmer, R. (2005).*Object-oriented systems analysis and design using UML*. McGraw Hill Higher Education.
- Kurbel, K. E. (2008). Information Systems Architecture. *The Making of Information Systems: Software Engineering and Management in a Globalized World*, 95-154.
- Hill, R., Hirsch, L., Lake, P., & Moshiri, S. (2012).*Guide to cloud computing: principles and practice*. Springer Science & Business Media.
- Furht, B., & Escalante, A. (2010). *Handbook of cloud computing* (Vol. 3). New York: Springer.
- Shroff, G. (2010). *Enterprise cloud computing: technology, architecture, applications*. Cambridge university press.
- Ahson, S. A., & Ilyas, M. (Eds.). (2010). *Cloud computing and software services: theory and techniques*. CRC Press.

Data mining and information retrieval

A short description of the course, topics:

Prerequisites:
The course requires basic knowledge in calculus, probability theory, and linear algebra. Knowledge of graphs and basic algorithms is an advantage.

The aim of the course is to provide a basic, but comprehensive introduction to data mining. By the end of the course, students will be able to build models, choose algorithms, implement and evaluate them.
Detailed Program and Class Schedule:

1. Motivations for data mining. Examples of application domains. Methodology of knowledge discovery in databases (KDD) and data mining (DM). Formulation of main problems of data mining.
2. Understanding data: preparation and exploration. Sampling.
3. Basics of classification. Concepts of training and prediction. Decision trees.
4. Models and algorithms for classification: k-NN, naïve-Bayes. Measuring quality and comparison of classification models.
5. Introduction to the WEKA data mining software. Classification with WEKA.
6. More models and algorithms for classification: neural networks, linear separation methods, support vector machine (SVM).
7. Basics of cluster analysis. Type of variables, measuring similarity and distances. Partitioning clustering algorithms, *k*-means, *k*-medoids.
8. Introduction to frequent itemset mining. The APRIORI algorithm. Applications for finding association rules.
9. Advanced classification methods: Bagging, boosting, AdaBoost.
10. Support Vector Machine. Kernel methods, graph kernels. Protein function prediction.

11. Dimensionality reduction by spectral methods, singular value decomposition, low-rank approximation.
12. Search engines, web information retrieval, PageRank and beyond.

Literature:

- Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Addison-Wesley, 2006.
- Jiawei Han és Micheline Kamber: Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann Publishers, 2006.
- T. Hastie, R. Tibshirani, J. H. Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag, 2001.

Design of Distributed Systems

A short description of the course, topics:

Students will be able to express and verify the properties of the distributed programs using formal methods, apply different ways to create advanced compositions of simple programs, and solutions for interesting and difficult problems in a distributed way.

Dining/drinking philosophers, formal specification of distributed problems, properties of distributed systems, safety and progress properties of distributed programs, verification of safety critical properties, program compositions from components with proven properties, computing the value of an associative function, message channels, pipelined networks programming exercises where the students apply the learned methods in the practice.

Literature:

- Misra, J.: A discipline of multiprogramming: programming theory for distributed applications (Springer, 2001)
- K. Mani Chandy and Jayadev Misra: Parallel Program Design: A Foundation (Addison-Wesley,
- Reading, MA, Reading, Mass., 1988)
- Lamport, L.: Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers (Addison-Wesley 2002)
- Schmidt, D., C. et al.: Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects (Wiley & Sons, 2000)

Formal semantics

A short description of the course, topics:

Introduction: motivation, approaches to semantics definitions
Translational semantics, attribute grammars and their applications
Denotational and operational semantics of expressions
Natural semantics of imperative statements
Structural operational semantics of imperative statements
Semantics of abort, nondeterministic and parallel execution
Denotational semantics of imperative statements
Domain and fixed point theory
Semantics of functional language elements
Modeling blocks and procedures
Modeling exceptions
Full abstraction

Literature:

- Hanne Riis Nielson and Flemming Nielson: Semantics with Applications - A Formal Introduction (John Wiley & Sons, 1992)
- Kenneth Slonneger and Barry L. Kurtz: Formal Syntax and Semantics of Programming Languages (Addison Wesley Longman, 1995)
- Glynn Winskel: The Formal Semantics of Programming Languages - An Introduction (Foundations of Computing Series, MIT Press, 1993)
- John C. Reynolds: Theories of Programming Languages (Cambridge University Press, 1998)

Functional languages

A short description of the course, topics:

Algebraic types, type classes.
Higher-order types, existential types.
Uniqueness typing.
Dynamics, generic programming.
Purely functional data structures.
Parallel and distributed programming.
Combinators, combinator libraries.
Monadic programming.
Interactive programs, Functional Reactive Programming.
Embedded domain-specific languages.

Literature:

- Koopman, P., Plasmeijer, R., van Eekelen, M., Smetsers, S. *Functional Programming in Clean*, 2002.
- Plasmeijer, R., van Eekelen, M., von Groningen, J. *Clean Language Report 2.2*, December 2011.
- Hudak, P. *The Haskell School of Expression, 1st Edition.* Cambridge University Press, February 2000.
- Gibbons, J., de Moor, O. *The Fun of Programming (Cornerstones of Computing).* Palgrave Macmillan, June 2005.
- Hutton, G. *Programming in Haskell.* Cambridge University Press, 2007.
- Thompson, S. *Haskell: The Craft of Functional Programming, 3rd Edition.* Addison-Wesley, June 2011.
- Marlow, S. *Parallel and Concurrent Programming in Haskell.* Proc. of the 4th Central European Functional Programming School, CEFP 2011, Budapest, Hungary, June 2011, Revised Selected Papers.

Recommended literature:

- O'Sullivan, B., Stewart, D., Goerzen, J. *Real World Haskell, 1st Edition.* O'Reilly Media, November 2008.
- Lipovaca, M. *Learn You a Haskell for Great Good! – A Beginner's Guide, 1st Edition.* No Starch Press, April 2011.

Interactive media design and development

A short description of the course, topics:

The course introduces Human–Computer Interaction (HCI) involving the study, planning, and design of the interaction between people (users) and computers.
Its aim is to understand the theoretical basics of Perception, Multimedia design, Information Visualization, Interaction Design, the Virtual Continuum, Serious Games, Tangible, Collaborative, Location-based, and Gesture-based technologies, etc.) and recent innovations in these areas.
Activities involve the exploration of emerging interactive technologies designed for demonstration, education, entertainment, navigation, narrative, support …etc. purposes and their variety of creative applications in different disciplines and user interest groups.
Students from different disciplines form groups to design and implement a specified innovative project that could well serve the basis of an industrial entrepreneurship.

Recommended literature:

- The Encyclopedia of Human-Computer Interaction, 2nd Ed. http://www.interaction-design.org/books/hci.html
- Journal of Virtual World Research: http://jvwresearch.org/
- Horizon Reports: http://www.nmc.org/horizon-project
- Papers submitted to conferences:
  - Museums and the Web: http://www.museumsandtheweb.com/
  - CHI: http://chi2013.acm.org/
  - iED: http://europe.immersiveeducation.org/events/ied-europe-summit-2012
  - DIS: http://www.dis2012.org/
  - ISMAR: http://ismar2011.vgtc.org/

Literature:

- C. Ware. Information Visualization - Perception for Design. (ed 3) 536 pp. Morgan Kaufmann. 2012. ISBN 978-0-12-381464-7
- Ed. Ioannis Deliyannis, Interactive Multimedia, ISBN 978-953-51-0224-3, Hard cover, 312 pages, Publisher: InTech, Chapters published March 07, 2012 under CC BY 3.0 license OpenAccess: http://www.intechopen.com/books/interactive-multimedia
- Lester Madden, Professional Augmented Reality Browsers for Smartphones: Programming for Junaio, Layar and Wikitude (Wrox Programmer to Programmer) ISBN-13: 978-1119992813
- L. Annetta and S. C. Bronack, (eds.), Serious Educational Game Assessment: Practical Methods and Models for Educational Games, Simulations and Virtual Worlds, 1–18. © 2011 Sense Publishers. ISBN: 978-94-6091-327-3 (paperback)
- The Functional Art: An Introduction to Information Graphics and Visualization (Peachpit/Pearson Education, 2012): http://www.thefunctionalart.com/ ISBN-13: 978-0321834737
- Ed. Xin-Xing Tang, Virtual Reality - Human Computer Interaction, ISBN 978-953-51-0721-7, Hard cover, 306 pages, Publisher: InTech, Chapters published September 05, 2012 under CC BY 3.0 license, OpenAccess: http://www.intechopen.com/books/virtual-reality-human-computer-interaction

Models of computation

A short description of the course, topics:

The aim of the course is to to provide a better understanding of the concept of computation and computational modelling by presenting different computational models. We discuss basic classical models as finite automata, pushdown automata, Turing machines and their variants (for example, register machines), partial recursive functions, random access machines, circuits, cellular automata, Petri nets. We also survey some emergent models of computation, as membrane systems and some models from DNA computing. We provide information on the computational power and efficiency of these constructs, examine their computational and descriptional complexity, and compare the different models with each other. We also discuss how these models can be used in solving theoretical and practical problems.

Literature:

- J. E. Savage, Brown University, Models of Computation, 1998. (www.cs.brown.edu/~jes/book/pdfs/ModelsOfComputation.pdf)
- M. Fernandez, Models of Computation: An Introduction to Computability Theory (Undergraduate Topics in Computer Science), Springer, 2009
- M. Sipser, Introduction to the Theory of Computation, 2nd edition, Thomson Course of Technology, 2006
- Gh. Paun, G. Rozenberg, A. Salomaa: DNA Computing – New Computing Paradigms. Springer, 1998
- Gh. Paun, Membrane Computing. An Introduction. Springer, 2002
- G. Rozenberg, T. Back, J.N. Kook (eds), Handbook of Natural Computing, Springer, 2012.

Recommended literature:

- J.E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages and Computation, 2nd edition, Addison-Wesley,2001
- J. Hromkovic, Theoretical Computer Science: Introduction to Automata, Computability, Complexity, Algorithmics, Randomization, Communication, and Cryptography (Texts in Theoretical Computer Science). Springer, 2007
- J. Hromkovic, Algorithmic Adventures: From Knowledge to Magic. Springer, 2009
- 4. Gh. Paun, G. Rozenberg, A. Salomaa (eds.), The Oxford Handbook of Membrane Computing. Oxford University Press, 2010.

Scalable enterprise applications

A short description of the course, topics:

The course presents some important application domains for distributed programming, with special regard to present software industry challenges and scientific computations. After the completion of the course the students will not only understand the theoretical issues of distributed computing, but they will

also be capable of designing and implementing distributed applications in general, and distributed object systems in particular. They will also learn common technologies used in the software industry. The following topics will be addressed (related technologies that can be used for illustration purposes are in parentheses).
Multi-tier application model: Modularization of large software systems, optimal use of distributed architectures in the design of the components (with respect to efficiency and high availability). Transactional applications backed by information systems. (Java EE, JDBC, JPA, JTA)
Remote Procedure Call: (Java RMI, EJB)
Message-based communication: (JMS, PVM/MPI)
Web-programming: Web-applications (Java servlet, JSP, JSF) , web-services (JAX-WS)
Component lookup: (JNDI, Jini).
Code mobility: (Java applet)
Grid systems: fulfilling high computational requirements.
Aspect-oriented programming: Used in the implementation of the above technologies. (AspectJ)

Literature:

- Jendrock, E., Ball, J., Carson, D., Evans, I., Fordin, S., Haase, K.: The Java EE 5 Tutorial, Third Edition (Addison-Wesley, 2007)
- http://java.sun.com/javaee/5/docs/tutorial/doc/
- Foster, I.: The Grid: Blueprint for a New Computing Infrastructure, 2nd Edition (Morgan Kaufmann, 2004)

Service Science

A short description of the course, topics:
Concepts and standards of Enterprise, Information and Software Architecture
Foundations of Service
Electronic Services
Service Innovation
Service Design
Which known methods and techniques are available to design services?
Service Semantics
Service Analytics
Service Optimization
Service Co-creation
Service Markets
Service Research
SOA – Service Oriented Architecture

Literature:

- Cardoso, J. (2015). *Fundamentals of Service Systems*. H. Fromm, S. Nickel, G. Satzger, R. Studer, & C. Weinhardt (Eds.). Springer.
- Qiu, R. G. (2014). *Service Science: The foundations of service engineering and management*. John Wiley & Sons.
- Newman, S. (2015). *Building Microservices*. " O'Reilly Media, Inc.".
- Familiar, B. (2015). *Microservices, IoT, and Azure*. Apress.
- Nadareishvili, I., Mitra, R., McLarty, M., & Amundsen, M. (2016). *Microservice Architecture: Aligning Principles, Practices, and Culture*. " O'Reilly Media, Inc.".
- Gadea, C., Trifan, M., Ionescu, D., & Ionescu, B. (2016, April). A reference architecture for real-time microservice API consumption. In *Proceedings of the 3rd Workshop on CrossCloud Infrastructures & Platforms* (p. 2). ACM.
- Thomas Erl, *Service-Oriented Architecture Concept*s, Technology and Design, 2005, Pearson Education
- Perks, Col., Beveridge, Tony, *Guide to enterprise IT architecture*, Springer-Verlag New York., ISBN 0-387-95132-6, 2003 .
- Daniel Minoli, *Enterprise Architecture A to Z Frameworks, Business Process Modeling, SOA, and Infrastructure Technology*, Auerbach Publications, Taylor & Francis Group, ISBN 978-0-8493-8517-9, 2008
- Martin Op ’t Land, Erik Proper, Maarten Waage, Jeroen Cloo, Claudia Steghuis, *Enterprise*

Architecture, Creating Value by Informed Governance, Springer-Verlag Berlin Heidelberg, ISBN 978-3-540-85231-5, 2009
- Marc Lankhorst et al., *Enterprise Architecture at Work*, 2005, Springer-Verlag Berlin Heidelberg, ISBN-10 3-540-24371-2
- Open Group, *TOGAF® Version 9.1,* http://pubs.opengroup.org/architecture/togaf9-doc/arch/
- Maglio, P. P., Kieliszewski, C. A., & Spohrer, J. C. (2010). *Handbook of service science* (p. 143). New York: Springer.
- Demirkan, H., Spohrer, J. C., & Krishna, V. (2011).*The science of service systems*. New York: Springer.

Software quality and testing

A short description of the course, topics:

Fundamentals of software testing
Fundamental test process
Testing throughout the software life cycle, Test levels,
Static techniques
Test design techniques
Specification-based or black-box techniques
State Transition Testing, Use case Testing
Structure-based or white-box techniques
Experience-based techniques
Test management
Risks and Testing
Tool support for testing
Case study

Literature:

- Dorothy Graham, Erik Van Veenendaal, Isabel Evans, Rex Black: Foundations of Software Testing, Cengage Learning EMEA, 2nd ed., 2008, ISBN-13: 978-1844809899
- Jamie L. Mitchell, Rex Black, Advanced Software Testing, Vol 3, Rocky Nook, 2nd ed., 2015, ISBN-13: 978-1-937538-64-4
- Graham Bath, Judy McKay, The Software Test Engineer's Handbook, Rocky Nook, 3rd ed, 2014, ISBN-13: 978-1-937538-44-6

Software Technology

A short description of the course, topics:

Purpose:
The course gives a broad overview of the process and methodologies of software development and its execution.
We cover all phases of development from requirements to maintenance and quality assurance with emphasize on architectural design.
The course tries to deliver a balanced mixture of theoretical knowledge and practical skills with currently used technologies.

Competencies delivered:
Students completing the class will understand software development process, its different strategies and methodologies.
They will be able to make sensible architectural decisions and plans well in advance using the acquired mixture of theoretical and hands-on skills.

Prerequisties:
- advanced knowledge of at least one object oriented programming language
- understanding of web technologies full stack (client, database, server...)
- (optional) project experience

Literature:

- R. C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall 2008.
- F. P. Brooks: The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley 1995.

- J. Humble, D. Farley: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley 2010.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
- M. R. Blaha, J. R. Rumbaugh: Object-Oriented Modeling and Design with UML, Pearson, 2004.
- L. Bass, P. Clements, R. Kazman: Software Architecture in Practice. 3rd ed. Addison-Wesley Professional, 2012.

Theory of programming

A short description of the course, topics:

Basic notions of programming. The syntax and semantics of nondeterministic programs. Partial and total correctness. Weakest precondition. The notion of loop invariant. Derivation rules of program constructs. Verification: a method for proving total correctness of deterministic and nondeterministic programs. Synthesizing correct sequential programs by using the derivation rules. The correctness of concurrent programs, verification rules of the new statements (parbegin/parend, await). Owicki-Gries method for proving the total correctness of parallel programs, deadlock freedom and interference freedom.

Literature:

- K. R. Apt, E.-R. Olderog. Verication of Sequential and Concurrent Program. Springer-Verlag, 1997.
- S. Owicki, D. Gries. An axiomatic proof technique for parallel programs. Acta Inf., 6, pp. 319-340, 1976
- E. W. Dijkstra. A Discipline of Programming. Prentice-Hall, Englewood Cliffs, New York, 1976.

Web engineering

A short description of the course, topics:

This curriculum introduces the students with the modern, state-of-the-art client and server side web technologies, methodologies of web engineering, the programming and design patterns, especially with the web service oriented architectures. By the end of the course the student has a global overview of the up-to-date web trends and technologies, and, with the help of them, is able to develop a web application and web information systems.

Introduction to Web Technologies and Web Engineering: specialties, characteristics, categories of web applications.
Web Architectures: multi-tier, data-centric architectures,
Requirement Analysis of Web Applications
Specialties of Large Enterprise and Small and Medium Enterprise Web Applications
Development Process of Web Applications
Model-Based Web Application Design and Development, WebML
Testing, Quality Management.
Design of Web 2.0 and Enterprise 2.0 Applications
Web Business Models
Web project management
Design of Mobile Web Applications
Semantic Web Applications, integration to Web Information Systems
Web Application Models, Cloud computing
Service Oriented Architectures, Web Information Systems

Literature:

- Kappel, G., Pröll, B., Reich, S., Retschitzegger W. (Eds.): Web Engineering: The Discipline of Systematic Development. John Wiley & Sons Inc., Chichester (2006).
- Mendes, E., Mosley, N. (Eds.): Web engineering. Springer-Verlag, Berlin (2005).
- Murugesan, S., Deshpande, Y. (Eds.): Web Engineering: Managing Diversity and Complexity of Web Application Development. LNCS 2016, Springer-Verlag, Berlin (2001).

Advanced cryptography

A short description of the course, topics:

The course have two main goals: discovering the mathematical background beyond several cryptographic constructions and introducing novel cryptographic primitives using interesting results from various topics of mathematics or computer science. For the first part, we present the necessary exact definitions, precise assumptions and rigorous proofs of security. For the second part, we present recent results, methods and its connections to cryptographic problems from finite fields to linear algebra.

Perfect and computational security, proofs by reduction, security definitions, pseudorandomness, message authentication codes, collision-resistant hash functions, one-way functions, cryptographic hardness assumptions, primality testing, factoring and computing discrete logarithms, arithmetics in finite fields and its applications, elliptic curve based cryptography, lattice based constructions, secure multiparty computation, secret sharing problems, applications for e-commerce.

Literature:

- Berlekamp, E.R.: Algebraic Coding Theory. McGraw Hill, 1968
- Huffman, W.C. –Pless, V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, 2003
- van Lint, J.H.: Introduction to coding theory. Springer Verlag, 1982
- McWilliams, F.J. – Sloane, M.J.A.: The theory of error-correcting codes. North-Holland, 1977
- Roman, S.: Coding and information theory. Springer Verlag, 1992
- Beutelspacher, A.: Cryptology. The Mathematical Association of America, 1994
- Brassard, G.: Modern cryptology. Springer Verlag, 1988
- van Tilborg: An introduction to cryptology. Kluiver Academic Publisher, 1988

Applied cryptography project seminar

A short description of the course:

The objective of the course is to develop and strengthen the ability to complete miniprojects, working in small groups (3 persons approx.). The practical aspects of the learned cryptographical solutions is emphasized, as well as focused team work concentrated on modeling and solving a security problem originated in a real, practical situation.

Literature:

- Berlekamp, E.R.: Algebraic Coding Theory. McGraw Hill, 1968
- Huffman, W.C. – Pless, V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, 2003
- van Lint, J.H.: Introduction to coding theory. Springer Verlag, 1982
- McWilliams, F.J. – Sloane, M.J.A.: The theory of error-correcting codes. North-Holland, 1977
- Roman, S.: Coding and information theory. Springer Verlag, 1992
- Beutelspacher, A.: Cryptology. The Mathematical Association of America, 1994
- Brassard, G.: Modern cryptology. Springer Verlag, 1988
- van Tilborg: An introduction to cryptology. Kluiver Academic Publisher, 1988

Cryptographic protocols

A short description of the course:

This course gives an overview of the basic building blocks used to engineer cryptographic protocols, and discusses in details the operation of mainstream cryptographic protocols used in wired and wireless computer networks. In particular, TLS and IPsec are covered, as well as security protocols in WiFi networks. We also study protocols used in emerging wireless networks, such as wireless sensor networks and RFID systems.

Basic concepts and crypto primitives
Block encryption modes
Message authentication and authenticated encryption
Key exchange protocols
Random number generation
Verification of key exchange protocols with ProVerif

Public Key Infrastructres
TLS
WiFi security
IPsec
Security protocols for wireless sensor networks
Secure routing and wormhole detection
RFID security and privacy

Literature:

- G. Schaefer, Security in Fixed and Wireless Networks, Wiley, 2004.
- J. Edney and W. A. Arbaugh, Real 802.11 Security: WiFi Protected Access and 802.11i, Addison-Wesley, 2003.
- L. Buttyán, JP. Hubaux, Security and Cooperation in Wireless Networks, Cambridge University Press, 2008.
- J. Lopez and Z-H. Zhou (eds), Wireless Sensor Network Security, IOS Press, 2008.
- A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

Economics of security and privacy

A short description of the course:

Introduction to security and microeconomics concepts, game theory primer
Incentive problems in information security
Basic defenses and security investments
Information gathering in security defense systems
Economics of privacy
Adoption of security solutions
Cyber-insurance and risk management
Advanced topics and additional discussion

Learning technologies

A short description of the course:

Educational Technology:
The informatics basics of educational technologies in building personal learning networks. Aggregation, filtering, sharing information using web technologies and resources. Teaching and learning in a networked society, choosing the best tools for specific learning situations, empowering the learner for maturing knowledge. Building a learning community of practice. Application of new pedagogies in teaching computational thinking, constructionism, project-based learning, exploratory learning, ... etc.

Competencies:
Motivating innovative thinking and critical thinking skills with respect to risks using ICT. Being sensitive to learners needs, learning styles and preferences, personalisation. Be aware of different learning theories, learning processes and the roles of learners and teachers. Being aware of the aims in teaching/using ICTs, developing different skills, transferring values and understandings on the effect of ICT on society.

Literature:

- Teachers' Educational Technology Guide:
  http://www.educatorstechnology.com/2012/08/teacher-educational-technology-guides.html
- Trend Report on OER:
  http://www.surf.nl/en/publicaties/Documents/trendrapport%20OER%202012_10042012%20%28ENGELS%20LR%29.pdf
- Horizon Reports:
  http://www.nmc.org/nmc-horizon/
- JISC: Effective practice in the digital age
  http://webarchive.nationalarchives.gov.uk/20140702233839/
  http://www.jisc.ac.uk/whatwedo/programmes/elearningpedagogy/practice.aspx
- Net Generation and Digital Native – Implication for HE
  https://www.heacademy.ac.uk/sites/default/files/next-generation-and-digital-natives.pdf

- Use of iPads in HE - Conference Proceedings
  https://www.academia.edu/6740633/Conference_Proceedings_1st_International_Conference_on_the_Use_of_iPads_in_Higher_Education_ihe2014

Recommended literature:

- Directory of Learning and Performance tools:
  http://c4lpt.co.uk/directory-of-learning-performance-tools/
- Sustainable Teacher Education:
  http://prezi.com/jli7ccsvvlo6/sustainable-innovation-in-teacher-education/

Programming in Erlang

A short description of the course:

Erlang is a dynamically typed functional programming language designed for building highly concurrent, distributed applications. However it is necessary to know the basic concepts of the functional programming paradigm to build an application like that. This course introduces the basic concepts of functional programming via implementation of sequential Erlang programs. Finally the actor model of Erlang is briefly introduced.

Properties of functional programming
Erlang history
Erlang VM
Erlang Terms
Modules and Functions in Erlang
Variables and Pattern Matching
Operation on different data types
Iterative evaluation: list comprehensions and recursive functions
Conditional Evaluation
Lambda-expressions
Dynamic constructs
Error handling
Records and maps
Macros
Binary
IO
Actor model and concurrency primitives

Network algorithms

A short description of the course:

Ad hoc networks do not use any extra infrastructure. The nodes of the network use a wireless communication interface and communicate directly and provide the routing necessary to deliver messages over multiple hops. We discuss medium access, routing algorithms, and methods dealing with the mobility of participants. The topics of the course: Modeling networks, capacity of wireless networks, topology control, routing, distributed localization, energy, dilation, congestion, mobility models.

Literature:

- H. Karl and A. Willig: Protocols and Architectures for Wireless Sensor Networks. Wiley, ISBN: 978-0-470-09510-2, 2005.
- Y. Wang: Topology Control for Wireless Sensor Networks. Book Chapter of Wireless Sensor Networks and Applications, Series: Signals and Communication Technology, edited by Li, Yingshu; Thai, My T.; Wu, Weili, Springer-Verlag, ISBN: 978-0-387-49591-0, 2008.
- XiuZhen Cheng, Xiao Huang, and Ding-Zhu Du (Editors): Ad Hoc Wireless Networking, Kluwer Academic Publishers, 2004.
- X.-Y. Li and Y. Wang: Wireless Sensor Networks and Computational Geometry. Book Chapter of Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, CRC Press, edited by Mohammad Ilyas et al., 2004.
- Ch. Scheideler: Overlay Networks for Wireless Systems. Book Chapter of Performance Analysis of Mobile and Ad Hoc Networks, Wireless Networks and Mobile Computing Series, Vol. 7, Nova Science Publishers, edited by Chansu Yu, 2007.