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 semesters 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.

<table>
<thead>
<tr>
<th>1st semester</th>
<th>2nd semester</th>
<th>3rd semester</th>
<th>4th semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data mining and information retrieval</td>
<td>Complex information systems (5 credits)</td>
<td>Advanced Java programming (5 credits)</td>
<td>Thesis Work (30 credits)</td>
</tr>
<tr>
<td>(5 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
<td>(30 credits)</td>
</tr>
<tr>
<td>Interactive media design and development</td>
<td>Formal semantics (3 credits)</td>
<td>Analysis of distributed systems (5 credits)</td>
<td>Software engineering lab 2. (5 credits)</td>
</tr>
<tr>
<td>(5 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
</tr>
<tr>
<td>Models of computation</td>
<td>Functional languages (5 credits)</td>
<td>Design of distributed systems (5 credits)</td>
<td></td>
</tr>
<tr>
<td>(5 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
</tr>
<tr>
<td>Preparation course for master studies and</td>
<td>Service science (5 credits)</td>
<td>Scalable enterprise applications (5 credits)</td>
<td></td>
</tr>
<tr>
<td>developing learning skills (0 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
</tr>
<tr>
<td>Theory of programming (5 credits)</td>
<td>Software quality and testing (5 credits)</td>
<td>Software engineering lab 1. (5 credits)</td>
<td></td>
</tr>
<tr>
<td>Web engineering (5 credits)</td>
<td>Software technology (5 credits)</td>
<td>(5 credits)</td>
<td>(5 credits)</td>
</tr>
<tr>
<td>elective course (5 credits)</td>
<td>elective course (2 credits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elective courses</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total credits</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 credits</td>
<td>30 credits</td>
<td>25 credits</td>
<td>35 credits</td>
</tr>
</tbody>
</table>
## 2. List of courses with their credit values and weekly hours

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

1 contact hour = 45 minutes

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

<table>
<thead>
<tr>
<th>Title of the course</th>
<th>Credits</th>
<th>RS</th>
<th>SO</th>
<th>LH/W</th>
<th>PH/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced cryptography L.</td>
<td>2</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Advanced cryptography Pr.</td>
<td>4</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Applied cryptography project seminar L.</td>
<td>2</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Applied cryptography project seminar Pr.</td>
<td>4</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Cryptographic protocols L.</td>
<td>2</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Cryptographic protocols Pr.</td>
<td>2</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Economics of security and privacy L.</td>
<td>4</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Learning technologies Pr.</td>
<td>3</td>
<td></td>
<td>Autumn, Spring</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Programming in Erlang</td>
<td>2</td>
<td></td>
<td>Spring</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Network Algorithms</td>
<td>5</td>
<td></td>
<td>Autumn</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
## 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
- Network programming: TCP and UDP; HTTP
- Program design principles and best practices
- Exceptions, assertions
- Logging and testing

**Literature:**


### 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:**


### 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
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.
5. Introduction to the WEKA data mining software. Classification with WEKA.
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:**
- Jiawei Han and Micheline Kamber: *Data Mining: Concepts and Techniques*, 2nd ed., Morgan Kaufmann Publishers, 2006.

### 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:**

### 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:**
- 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:**

**Recommended literature:**

**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 Continum, 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](http://www.interaction-design.org/books/hci.html)
- Horizon Reports: [http://www.nmc.org/horizon-project](http://www.nmc.org/horizon-project)
- Papers submitted to conferences:
  - iED: [http://europe.immersiveeducationeducation.org/events/ied-europe-summit-2012](http://europe.immersiveeducationeducation.org/events/ied-europe-summit-2012)
### Literature:

### 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:
- M. Fernandez, Models of Computation: An Introduction to Computability Theory (Undergraduate Topics in Computer Science), Springer, 2009

### Recommended literature:

### 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:**


### 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:**

- Martin Op ’t Land, Erik Proper, Maarten Waage, Jeroen Cloo, Claudia Steghuis, *Enterprise
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:

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.

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

Literature:
- 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.

### Theory of programming

**A short description of the course, topics:**

**Literature:**

### 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:**
**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:**

- van Lint, J.H.: Introduction to coding theory. Springer Verlag, 1982
- Roman, S.: Coding and information theory. Springer Verlag, 1992
- 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:**

- van Lint, J.H.: Introduction to coding theory. Springer Verlag, 1982
- Roman, S.: Coding and information theory. Springer Verlag, 1992
- 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 Infrastructures
TLS
WiFi security
IPsec
Security protocols for wireless sensor networks
Secure routing and wormhole detection
RFID security and privacy

Literature:

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:
- Horizon Reports: http://www.nmc.org/nmc-horizon/
Use of iPads in HE - Conference Proceedings

Recommended literature:
- Directory of Learning and Performance tools: http://c4lpt.co.uk/directory-of-learning-performance-tools/

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: