Curriculum for the Master’s Programme in Computer Science (IT)

Aalborg University
September 2017
Preface:
Pursuant to Act 261 of March 18, 2015 on Universities (the University Act) with subsequent changes, the following curriculum for the Master's programme in Computer Science (IT) is stipulated. The programme also follows the Joint Programme Regulations and the Examination Policies and Procedures for The Technical Faculty of IT and Design, The Faculty of Engineering and Science, and The Faculty of Medicine.
# Table of contents

CHAPTER 1: LEGAL BASIS OF THE CURRICULUM, ETC. ........................................................................... 4

1.1 Basis in ministerial orders ........................................................................................................ 4
1.2 Faculty affiliation ....................................................................................................................... 4
1.3 Board of Studies affiliation ...................................................................................................... 4
1.4 Body of external examiners ...................................................................................................... 4

CHAPTER 2: ADMISSION, DEGREE DESIGNATION, PROGRAMME DURATION AND COMPETENCE PROFILE .......................................................................................... 4

2.1 Admission ................................................................................................................................ 4
2.2 Degree designation in Danish and English ............................................................................... 4
2.3 The programme’s specification in ECTS credits ...................................................................... 5
2.4 Competence profile on the diploma ......................................................................................... 5
2.5 Competence profile of the programme: .................................................................................. 5

CHAPTER 3: CONTENT AND ORGANIZATION OF THE PROGRAMME ....................................................... 6

1. semester, IT7 ............................................................................................................................. 9
2. semester, IT8 ............................................................................................................................ 20
3. semester, IT9 ........................................................................................................................... 31
4. semester, IT10 .......................................................................................................................... 39

CHAPTER 4: ENTRY INTO FORCE, INTERIM PROVISIONS AND REVISION ........................................ 40

CHAPTER 5: OTHER PROVISIONS ...................................................................................................... 40

5.1 Rules concerning written work, including the Master’s thesis ............................................. 40
5.2 Rules concerning credit transfer (merit), including the possibility for choice of modules that are part of another programme at a university in Denmark or abroad ......................................................... 40
5.3 Rules for examinations .......................................................................................................... 40
5.4 Exemption .............................................................................................................................. 40
Chapter 1: Legal Basis of the Curriculum, etc.

1.1 Basis in ministerial orders
The Master’s programme in Computer Science (IT) is organized in accordance the Ministry of Higher Education and Science’s Order no. 1328 of November 15, 2016 on Bachelor’s and Master’s Programmes at Universities (the Ministerial Order of the Study Programmes) and Ministerial Order no. 1062 of June 30, 2016 on University Examinations (the Examination Order). Further reference is made to Ministerial Order no. 111 of January 30, 2017 (the Admission Order) and Ministerial Order no. 114 of February 3, 2015 (the Grading Scale Order) with subsequent changes.

1.2 Faculty affiliation
The Master’s programme falls under The Technical Faculty of IT and Design, Aalborg University.

1.3 Board of Studies affiliation
The Master’s programme falls under the Board of Studies for Computer Science.

1.4 Body of External Examiners
The Master’s programme is associated with the Body of External Examiners for Computer Science.

Chapter 2: Admission, Degree Designation, Programme Duration and Competence Profile

2.1 Admission

Applicants with a legal claim to admission (retnkrav):
Applicants with one of the following degrees are entitled to admission:

- Bachelor’s degree in Information Technology with Specialisation in Technology from Aalborg University

Applicants without legal claim to admission:
Students with Bachelor’s degree in Computer Science or Software Engineering may, upon application to the Board of Studies, be admitted after a specific academic assessment if the applicant is deemed to have comparable educational prerequisites. The University can stipulate requirements concerning conducting additional exams prior to the start of study.

All students applying must document English language qualifications comparable to an ‘English B level’ in the Danish upper secondary school (minimum average grade 02).

2.2 Degree designation in Danish and English
The Master’s programme entitles the graduate to the designation cand.scient. (candidatus/candidata scientiarum) i datalogi (it). The English designation is: Master of Science (MSc) in Computer Science (IT).
2.3 The programme’s specification in ECTS credits
The Master’s programme is a 2-year, research-based, full-time study programme. The programme is set to 120 ECTS credits.

2.4 Competence profile on the diploma
The following competence profile will appear on the diploma:

A Candidatus graduate has the following competency profile:
A Candidatus graduate has competencies that have been acquired via a course of study that has taken place in a research environment.

A Candidatus graduate is qualified for employment on the labour market on the basis of his or her academic discipline as well as for further research (PhD programmes). A Candidatus graduate has, compared to a Bachelor, developed his or her academic knowledge and independence so as to be able to apply scientific theory and method on an independent basis within both an academic and a professional context.

2.5 Competence profile of the programme:
The graduate of the Master’s programme:

Knowledge
- has knowledge in computer science that, in selected areas, is based on the highest level of international research in the subject area.
- can understand and, on a scientific basis, reflect over challenges in computer science and identify solutions to scientific and engineering problems

Skills
- are proficient in scientific methods and tools and general skills related to computer science
- can evaluate and select among the theories, methods, tools and general skills and, on a sound scientific and engineering basis, advance new analyses and solutions
- can communicate research-based knowledge and discuss professional, scientific and engineering problems with both peers and non-specialists

Competences
- can manage work and development situations that are complex, unpredictable and require new solutions.
- can independently initiate and implement discipline-specific and interdisciplinary cooperation and assume professional responsibility.
- can independently take responsibility for own professional development and specialization.
Chapter 3: Content and Organization of the Programme

The programme is structured in modules and organized as a problem-based study. A module is a programme element or a group of programme elements, which aims to give students a set of professional skills within a fixed time frame specified in ECTS credits, and concluding with one or more examinations within specific exam periods. Examinations are defined in the curriculum.

The programme is based on a combination of academic, problem-oriented and interdisciplinary approaches and organized based on the following work and evaluation methods that combine skills and reflection:

- lectures
- classroom instruction
- project work
- workshops
- exercises (individually and in groups)
- teacher feedback
- reflection
- portfolio work

Overview of the programme:

All modules are assessed through individual grading according to the 7-point scale or Pass/Fail. All modules are assessed by external examination (external grading) or internal examination (internal grading or by assessment by the supervisor only).

<table>
<thead>
<tr>
<th>Semester</th>
<th>Module</th>
<th>ECTS</th>
<th>Assessment</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (IT7)</td>
<td>From Reality to Models</td>
<td>15</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Programming Paradigms</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td><strong>Two of the following:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computability and Complexity</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td>Advanced Topics in Databases</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Machine Intelligence</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Advanced Topics in Human-Computer Interaction</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Data-Intensive Systems</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Advanced Topics in Distributed Systems</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Real Time Systems</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Web Engineering</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Web Intelligence</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td>2 (IT8)</td>
<td>From Models to Reality</td>
<td>15</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td>Software Engineering</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td><strong>Two of the following:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Advanced Algorithms</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Mobile Software Technology</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Advanced Programming</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td>Course</td>
<td>Credits</td>
<td>Scale</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>---------</td>
<td>---------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>Software Innovation</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
<td></td>
</tr>
<tr>
<td>Languages and Compilers</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>Principles of Operation Systems and Concurrency</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>Advanced Topics in Machine Intelligence</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
<td></td>
</tr>
<tr>
<td>Test and Verification</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
<td></td>
</tr>
<tr>
<td>Advanced Topics in Modelling and Verification</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
<td></td>
</tr>
<tr>
<td>3 (IT9) Pre-Specialisation in Computer Science</td>
<td>20</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>Entrepreneurship</td>
<td>5</td>
<td>Pass/Fail</td>
<td>Internal</td>
<td></td>
</tr>
<tr>
<td><strong>One of the following Specialisation Courses (SC)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC in Database Technology</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>SC in Distributed Systems</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>SC in Human-Computer Interaction</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>SC in Semantic and Verification</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>SC in Machine Intelligence</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>SC in Systems Development</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>SC in Programming Technology</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>4 (IT10) Master’s Thesis</td>
<td>30</td>
<td>7-point scale</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>120</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Problem Based Learning:**
A compulsory course in Problem Based Learning (PBL) is offered as an integrated part of the project module to students not acquainted with PBL at Aalborg University.
Tracks
After having been accepted to the programme, students are free to choose between the elective courses at IT7. Choosing one of the tracks below, i.e. Information Technology, Machine Intelligence, Data Engineering or Embedded Systems will give the student a coherent profile in the given track. Students are required to choose tracks at IT8. In addition to the tracks listed in the scheme below, it is also possible to create other combinations after a personal consultation with the Study Board. Students with a Bachelor’s degree in Information Technology will be enrolled at the IT-track.

<table>
<thead>
<tr>
<th>Semester</th>
<th>Mandatory course</th>
<th>Track IT: Information Technology</th>
<th>Track MI: Machine Intelligence</th>
<th>Track DE: Data Engineering</th>
<th>Track ES: Embedded Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS/IT7</td>
<td>Programming Paradigms</td>
<td>Computability and Complexity and one of the courses Advanced Topics in Databases Machine Intelligence Advanced Topics in Human-Computer Interaction</td>
<td>Machine Intelligence Web Intelligence</td>
<td>Machine Intelligence and one of the courses Advanced Topics in Databases Data-Intensive Systems</td>
<td>Advanced Topics in Distributed Systems Real Time Systems</td>
</tr>
<tr>
<td>CS/IT8</td>
<td>Software Engineering</td>
<td>Languages and Compilers and one of the courses Advanced Algorithms Advanced Programming Principles of Operation Systems and Concurrency</td>
<td>Advanced Algorithms Advanced Topics in Machine Intelligence</td>
<td>Advanced Algorithms Advanced Programming</td>
<td>Test and Verification and one of the courses Advanced Topics in Modelling and Verification Advanced Programming</td>
</tr>
<tr>
<td>CS/IT9</td>
<td>Entrepreneurship Specialisation course (one of 7 courses)</td>
<td>Pre-Specialisation in Computer Science</td>
<td>Pre-Specialisation in Computer Science</td>
<td>Pre-Specialisation in Computer Science</td>
<td>Pre-Specialisation in Computer Science</td>
</tr>
<tr>
<td>CS/IT10</td>
<td></td>
<td>Master’s Thesis</td>
<td>Master’s Thesis</td>
<td>Master’s Thesis</td>
<td>Master’s Thesis</td>
</tr>
</tbody>
</table>

Mandatory courses in bold.
1. semester, IT7

Title: From Reality to Models (Fra virkelighed til modeller)

15 ECTS (Project)

Purpose: The student should learn how using advanced models can help identifying and solving a problem of application or research in computer science and how analysis of problems can contribute to theory building in computer science.

Reason: A graduate in computer science should be able to apply advanced theory and model building to identify and contribute to solutions of problems in the area of computer science and in this context to reflect on the relevant theories and building of models.

Objective:

Knowledge:
Students who complete the module should be able to:
- explain concepts, findings and theories in an advanced area of computer science

Skills:
Students who complete the module should be able to:
- apply knowledge from a theory in computer science to select and argue for a model within an advanced computer science field
- from such modeling provide a model of a computer science problem and apply this model to understand the problem

Competences:
Students who complete the module should be able to:
- identify a problem given an open challenge in research or application of computer science
- contribute to solving the problem using their own modeling based on computer science theories
- analyze and evaluate the resulting contribution to the solution
- analyze and evaluate the applications of relevant computer science models to solve this problem

Type of instruction: Project work including
- an analysis and definition of a computer science problem
- establishment of an advanced computer science model that contributes to solving the problem
The project may include complete or partial implementation of a solution in the form of running software.

As an integrated part of the project work, the student must participate in the problem-based learning and project management workshop. Approved participation is required to register for the project exam.

Exam format: Oral exam, based on project
Assessment: Internal assessment, 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Titel: Problem Based Learning (PBL) at Aalborg University (Problembaseret læring på Aalborg Universitet)
Prerequisites: None, but the course is compulsory for students not acquainted with the PBL model at Aalborg University
Objectives: After completion of the course the student should

Knowledge:
- know how to describe in own words some of the fundamental principles of Problem Based Learning (PBL) as implemented in the Aalborg PBL model at the Faculty of Engineering and Science
- know how to identify similarities and differences between the Aalborg PBL study environment and previous study environments, incl. strengths and weaknesses in both environments

Skills:
- be able to structure project management activities based on a well-formulated problem formulation
- be able to assess project documentation based on scientific codes of conduct

Competences:
- to plan for effective collaborative learning in an intercultural environment and manage group conflicts
- be able to reflect on, plan and manage a study project in a PBL learning environment

Type of Instructions: Lectures, discussions and group work. The course will take place during three afternoons.

Assessment: Internal assessment during the course/class participation according to the rules in the Examination Policies and Procedures, Addendum to the Joint Programme Regulations of the Technical Faculty og IT and Design, The
Faculty of Engineering and Science, and The Faculty of Medicine, Aalborg University. In this case the assessment is primarily based on the oral performance during the course. This means that the student has to be active during the course time and participate in discussions. The course is an integrated part of the project and a precondition for participation in the project examination for those who are not acquainted with the Aalborg PBL model. Consequently, no diploma will be issued for the course nor will it appear on the academic transcripts. The assessment is pass/fail.

Evaluation criteria:
As stated in the Joint Programme Regulations, it is a precondition for students who have not studied the PBL model at Aalborg University that they have passed the course in Problem Based Learning (PBL) at Aalborg University prior to the project examination.

Title: Programming Paradigms
(Programmingssparadigmer)

5 ECTS (course)

Prerequisites:
Knowledge of Imperative Programming and Object Oriented Programming

Objective:
Knowledge:
The student will acquire knowledge of programming paradigms that are alternative and complementary to the imperative paradigm and the object oriented paradigm. Furthermore, students should acquire knowledge on advanced topics in design, implementation and use of programming languages and environments, including
• function-oriented programming
• programming language with dynamic types
• programming techniques in the field of one or more of the four main paradigms: the function-oriented, the imperative, the object-oriented and the logic programming paradigms

Skills:
The student must achieve the following skills:
• be able to classify and explain the structures in programming languages in relation to the paradigms
• be able to relate language constructions that support different paradigms
• be able to assess the strengths and weaknesses of each paradigm in relation to specific tasks

Competencies:
The student should be able to use paradigmatic constructions in smaller programmes.

Teaching form: Course

Exam form: Oral or written exam

Assessment: External assessment according to 7-point scale
Title: Computability and Complexity (Beregnelighed og kompleksitet)

5 ECTS (Course)

Prerequisites: The module adds to knowledge obtained in Syntax and Semantics

Objective: Knowledge:
Students should achieve knowledge on the following theories and methods:

Computability:
- deterministic and nondeterministic Turing machines; decidable and recognizable languages and their properties: Church-Turing thesis
- acceptance problem for Turing machines; other undecidable problems for Turing machines; reductions and their properties

Complexity theory:
- time complexity of deterministic and nondeterministic Turing machines; time complexity classes, polynomial reductions and their uses; NP-completeness; satisfiability problem (SAT); other NP-complete problems
- space complexity of deterministic and nondeterministic Turing machines; space complexity classes, the relationship between time and space complexity

The course will also involve one or more advanced topics that can be e.g. other models of computation, other results on undecidability or results about further complexity classes.

Skills:
Students should achieve the following skills:
- the ability to explain course concepts precisely using the terminology and notations of the discipline for important achievements in the theory of computability and computational complexity, and how and to what extent these results can be used to classify computational problems
- the ability to make use of the necessary writing skills in these contexts
Competencies:
The student should be able to apply concepts and techniques from the theory of computability and computational complexity for the analysis of computational problems

Teaching form: Course
Exam form: Oral or written exam
Assessment: External assessment, 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Advanced Topics in Databases
(Avancerede emner indenfor databaser)

5 ECTS (course)

Objective: Knowledge:
The student will acquire knowledge on the following topics in advanced databases:
• concepts and techniques in multidimensional databases, such as data warehousing, On-Line Analytical Processing, and Data Mining
• concepts and techniques in spatial and spatiotemporal databases, including indexing and processing of queries
• concepts and techniques of complex data in databases, such as XML, Semantic Web, etc.

There will also be one or more optional subjects within data-intensive systems, including (but not limited to):
• concepts and techniques in temporal databases

Skills:
The student must achieve the following skills:
• able to explain concepts and techniques in advanced databases
• able to identify and discuss relevant concepts and techniques for a given problem in advanced databases
• able to apply relevant concepts and techniques for a given problem in advanced databases

Competencies:
The student must be able to apply concepts and techniques from advanced databases, including the design and implementation of advanced databases

Teaching form: Course
Exam form: Oral or written exam
Assessment: Internal assessment according to 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Machine Intelligence (Maskinintelligens)

5 ECTS (course)

Prerequisites: Knowledge of discrete mathematics, algorithms and data structures

Objective: Knowledge:
Students should acquire knowledge on:
• basic techniques and methods in machine intelligence including their theoretical foundations and practical applications
• the use of correct technical notation and terminology

Skills:
The student should achieve the following skills:
• use basic techniques presented in the course to solve a specific problem
• use correct technical notation and terminology in both writing and speech
• be able to explain the key principles and algorithms presented in this course

Competencies:
The student should be able to evaluate and compare different machine intelligence techniques and methods based on a specific problem.

Teaching form: Course

Exam form: Oral or written exam

Assessment: Internal assessment, 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Advanced Topics in Human-Computer Interaction (Avancerede emner indenfor menneske-maskine interaktion)

5 ECTS

Objective: Knowledge:
The student will gain knowledge in advanced topics in human-machine interaction in theory or practice. The items may include, but are not limited to:
• concepts, methods and techniques in advanced interaction design
• concepts, methods and techniques in advanced usability evaluation

Skills:
The student must achieve the following skills within the course subject matter:
- able to explain issues, theory, methods, results and conclusions in an accurately and profound way
- be able to apply theories and methods to solve a specific problem
- have a critical approach to theories and methods in human-computer interaction

**Competencies:**
The student should be able to apply concepts, techniques and methods to understand a given problem and to design and evaluate a practical system.

**Teaching form:** Course
**Exam form:** Oral or written exam
**Assessment:** Internal assessment according to 7-point scale
**Evaluation criteria:** Are stated in the Joint Programme Regulations

---

**Title:** Data-Intensive Systems
(Data-intensive systemer)

5 ECTS

**Prerequisites:** Knowledge of database systems

**Objective:**

**Knowledge:**
Students should achieve knowledge on the following topics in data-intensive systems:
- concepts and techniques for analyzing large data volumes, such as data warehousing, On-Line Analytical Processing, and Data Mining
- concepts and techniques for handling spatio-temporal data, including indexing and processing of queries
- concepts and techniques for scalability for data-intensive systems, e.g., cloud computing

Topics will typically be exemplified by Internet-related application, such as web analytics, spatial web, and the like.

There will also be one or more optional subjects within data-intensive systems, including but not limited to:
- concepts and techniques for managing web-related data such as XML, Semantic Web, and Web2.0 data
- concepts and techniques for search engines

**Skills:**
The student should achieve the following skills:
- be able to explain concepts and techniques in data-intensive systems
be able to select and apply relevant concepts and techniques for a given problem in data-intensive systems

**Competences:**
The student should be able to apply concepts and techniques from data-intensive systems, including design and implementation of data-intensive systems.

**Teaching form:** Course

**Exam form:** Oral or written exam

**Assessment:** Internal assessment, 7-point scale

**Evaluation criteria:** Are stated in the Joint Programme Regulations

---

**Title:** Advanced Topics in Distributed Systems
*(Avancerede emner indenfor distribuerede systemer)*

5 ECTS (course)

**Objective:**

**Knowledge:** The student will gain knowledge of advanced theories and methods in distributed systems:
- advanced infrastructures and applications for example, grid, cloud, peer-to-peer, or parallel / multi-core systems
- System and Network software for embedded systems
- examples of distributed embedded systems, such as ad-hoc sensor networks, home automation
- Distributed algorithms such as algorithms for mutual exclusion, selection, consensus, replication and fault tolerance
- paradigms of programming
- techniques for analysis, such as monitoring, testing, verification, and benchmarking

**Skills:** The student must achieve the following skills:
- able to assess and explain precisely how and to what extent the results presented can be used using the appropriate subject terminology and notation,
- use appropriate writing skills in these contexts

**Competencies:** The student must be able to apply concepts and techniques from distributed systems, and to design and analyze distributed and embedded systems.

**Teaching form:** Course
Exam form: Oral or written exam
Assessment: Internal assessment according to 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Real-Time Systems (Tidstro software)

5 ECTS (course)

Prerequisites: Knowledge of computer architecture and principles of operating systems and parallelism

Objective:

Knowledge:
- Students should achieve knowledge on the following theories and methods:
  - design: tasks, temporal scopes, file management strategies, mode, change, synchronous and asynchronous interaction
  - Analysis: scheduling, response time analysis, modeling, verification and validation, priority protocols, hardware limitations
  - implementation: programming language with support for realtime programming, hardware abstraction, and system near programming, synchronization, atomicity, deadlocks, error handling, communication

The course will also involve one or more advanced topics that can be e.g. other principles for implementing or reasoning about real-time systems.

Skills:
The student should achieve the following skills:
- the ability to explain course concepts precisely using the terminology of the discipline and notation for overall design, analysis and implementation of simple real-time software systems
- the ability to apply relevant techniques to determine the possibility of scheduling a simple real-time application

Competencies:
The student should, in the synthesis of the concepts and techniques of the discipline:
- be able to design, analyze and implement a simple (embedded) real-time application
- be able to acquire new knowledge about the design, analysis and implementation of real-time systems

Teaching form: Course
Exam form: Oral or written exam.
Assessment: Internal assessment, 7-point scale
Title: Web Engineering (Web engineering)

5 ECTS (course)

Objective:
Knowledge:
The student should gain knowledge of developing web applications:
- types of web applications and their use (eg. Data-intensive, service-oriented collaboration, integration, social)
- types of web technologies
- methods for developing web applications
  - Requirements, design, implementation and testing techniques
  - Patterns for web applications
  - Development of web applications
- advanced topics in web development, for example:
  - Service-oriented architecture
  - Semantic web
  - Rich Internet Applications
  - New trends

Skills:
The student should achieve the following skills:
- demonstrate knowledge of web applications, web development and web architecture
- perform model-based analysis of web applications
- apply methods for developing web applications, including requirements, design, implementation and testing techniques

Competencies:
The student should be able to apply concepts and techniques from Web engineering, including web applications and development and architecture, requirements, design, implementation, and testing techniques.

Teaching form: Course

Exam form: Oral or written exam

Assessment: Internal assessment, 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations
Title: Web Intelligence
(Web Intelligence)

5 ECTS (course)

Prerequisites: Knowledge of discrete mathematics, algorithms and data structures

Objectives:

Knowledge:
Students should achieve knowledge and skills within web intelligence techniques, such as:
- Application of web intelligence techniques
- Web agents and web services
- Web information retrieval
- Web navigation support
- Recommender systems
- Intelligence for social web
- Presentation of knowledge and semantic web
- User modelling, adaptation and personification
- Computational natural language processing for web

Skills:
The student should achieve the following skills:
- Demonstrate knowledge about web intelligence methods and techniques
- Chose relevant concepts and techniques within a given web system problem
- Use correct notation and terminology for web intelligence.

Competences:
The student should be able to apply web intelligence methods and techniques, including design and implementation of web systems.

Teaching form: Course

Exam form: Oral or written exam

Assessment: Internal assessment, 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations
2. semester, IT8

Title: From Models to Reality (Fra modeller til virkelighed)

15 ECTS (project)

Prerequisites: The module adds to knowledge obtained in IT7 and the mandatory course modules on IT8

Purpose: The student should gain further insight into how using computer science models can help identifying and solving a problem with computer science research and application.

Reason: A graduate in computer science should be able to apply theory and model building to identify and to contribute to solutions of a computer science problem and in this context to reflect on the theories and building of models.

Objective:

Knowledge: Students who complete the module should be able to:
- explain the concepts, findings and theories in an advanced area of computer science

Skills: Students who complete the module should be able to:
- apply knowledge from a theory in computer science to select and argue for a model within an advanced computer science field
- from such modeling provide a model of a computer science problem and apply this model to understand the problem

Competences: Students who complete the module should be able to:
- identify a problem in research or application of computer science
- contribute to solving the problem using their own modeling based on theories
- analyze and evaluate the resulting contribution to the solution
- analyze and evaluate application of relevant models to solve the problem

Type of instruction: Project work including
- an analysis and description of a computer science problem
• establishment of an advanced computer science model that contributes to solving the problem

The project may include complete or partial implementation of a solution in the form of running software.

Exam format: Oral exam, based on project
Assessment: External assessment, 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Software Engineering (Software engineering)
5 ECTS (Course)

Objectives: Knowledge:
The student should gain knowledge of leading paradigms (e.g. traditional and agile) in professional development of software. The student should also gain knowledge of theories, methods and techniques involved in these paradigms (e.g. process modelling, management of requirements, design, project management, testing, process improvement) as well as an overview of theory of science for software engineering.

Skills:
The student should achieve the following skills:
• the ability to explain course concepts precisely using the terminology of the discipline, and be able to distinguish between and compare the software engineering paradigms.
• Be able to explain accurately and using terminology of the discipline, the theories, methods and techniques of software engineering paradigms and their application in the professional development of software intensive systems.

Competencies:
The student should be able to select, justify and use appropriate paradigms, theories, methods and techniques in their own development contexts.

Teaching form: Course
Exam form: Oral or written exam
Assessment: External assessment, 7-point scale.
Evaluation criteria: Are stated in the Joint Programme Regulations
Title: Advanced Algorithms (Avancerede algoritmer)

5 ECTS (Course)

Prerequisites: The module adds to knowledge obtained in Computability and Complexity and knowledge of algorithms and data structures, principles of operating systems and parallel systems

Objective: Knowledge:
Students should achieve knowledge on the following theories and methods:
- algorithm design techniques such as divide-and-conquer, greedy algorithms, dynamic programming, back-tracking, Branch-and-bound algorithms and plane-sweep algorithms
- algorithm analysis techniques such as recurrences, amortized analysis, analysis of the expected complexity and experimentation with algorithms
- a set of core algorithms and data structures for solving problems from different computer science areas: algorithms for external memory, multi-threaded algorithms, text search, advanced graph algorithms, heuristic search and computational geometry

There will also be one or more optional subjects in advanced algorithms including, but not limited to: approximate algorithms, randomized algorithms, linear programming and number theoretic algorithms such as cryptosystems.

Skills:
The student should achieve the following skills:
- ability to explain the principles behind the main algorithm design and algorithm analysis techniques
- select and apply the algorithm design and algorithm analysis techniques for a given problem
- recognize a number of problems from different computer science fields and select the most appropriate algorithms and data structures for solving them
- Argue about the correctness of selected algorithms, in particular, selected dynamic-programming, greedy, and approximation algorithms

Competencies:
When faced with a non-standard computer science problem, the student should be able to:
- develop efficient algorithms and data structures for solving the problem
- analyze the developed algorithms
Teaching form: Course

Exam form: Oral or written exam

Assessment: Internal assessment, 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Mobile Software Technology (Mobil softwareteknologi)

5 ECTS (Course)

Prerequisites: Knowledge and skills in object oriented programming, algorithms and data structures, database systems and design and evaluation of user interfaces.

Objective: Knowledge:
The student should gain knowledge in the following topics within development of mobile software:
- issues relating to architecture and programming of mobile software, such as stand-alone, client / server and peer to peer
- positioning and tracking both indoors and outdoors
- mobile services and location-based services
- mobile databases
- interaction design for mobile technologies
- usability and user experience evaluation of mobile technologies

In addition, the following items may be included in the course:
- indoor / outdoor integration
- Middleware platforms for mobile services
- design sketching for mobile technologies
- paper prototype development for mobile technologies
- lab. vs. field-based evaluation of mobile technologies

Skills:
The student should be able to:
- design software architectures for mobile applications
• use positioning and tracking techniques in various indoor and outdoor scenarios

• explain the principles for mobile databases

• explain the principles of moving object databases

• produce and refine the interaction design for mobile systems, services or devices

• evaluate the quality of an interaction design empirical

Competences:
The student should learn typical technologies and interaction design principles for mobile software systems and be able to use these technologies and principles in various mobile application scenarios.

Teaching form: Course
Exam form: Oral or written exam
Assessment: Internal assessment, 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Advanced Programming (Avanceret programmering)
5 ECTS (Course)

Prerequisites: Programming experience and knowledge of imperative and object-oriented programming. Knowledge about language design and compiler construction.

Objective: Knowledge:
The student should gain knowledge of advanced programming technologies and techniques, including elements of the programming language that supports these techniques. The course will focus on both new trends in programming, and on classic advanced themes. Possible topics include:

• advanced libraries
• library design
• syntactic abstraction (macros) and language extensions
• declarative programming
• generic programming
• concurrent, parallel and distributed programming
• reactive programming
• typed and typeless programming
• scripting
• module concepts
• different hardware platforms
• resource
• optimizations
• performance studies

Skills:
Students should achieve skills in selecting appropriate programming tools for a given task. The student should be able to write correct, efficient and maintainable programs. The student should be able to assess use of resources and to perform optimizations and performance studies.

Competencies:
The student should be able to solve advanced programming tasks.

Teaching form: Course
Exam form: Oral or written exam
Assessment: Internal assessment according to 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Software Innovation (Softwareinnovation)

5 ECTS (course)

Prerequisites: The module adds to knowledge obtained in the previous semester.

Objectives: In this context, software innovation implies the wide definition of innovation related to software development. Emphasis is on innovation, products and processes but also leadership of innovative work and personal and organizational prerequisites for innovation are included in the course.

Knowledge:
The student will acquire knowledge on the following theories and methods:

Software innovation theory:
• central theories about innovation and innovation processes
• personal and organizational conditions for innovation
• theories of software innovation

Innovation methods:
• general methods and techniques to support innovation
• specific methods and techniques for software innovation

Innovation practice:
• experience with methods and techniques in creative and innovative processes
- assessment of strengths and weaknesses of the methods and techniques for creative and innovative processes for software development

Skills:
The student must achieve the following skills:
- able to explain theories accurately using professional concepts
- able to explain approaches to selection and management of innovative processes in software development
- able to discuss personal and organizational prerequisites for software innovation
- use own experience to explain and discuss tools and techniques supporting software innovation

Competencies:
The student should be able to assess the innovative potential of a software product or software-supported process.

Teaching form: Course
Exam form: Oral or written exam
Assessment: Internal assessment according to 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: **Languages and Compilers**
(Sprog og oversættere)

5 ECTS (Course)

Prerequisites:
Programming experience and knowledge of imperative and object-oriented programming

Objective:
Knowledge:
Students should acquire knowledge about essential principles of programming languages, and understand techniques for description and translation of languages in general, including:
- Abstraction principle, control and data structures, block structure and scope concept, parameter mechanisms and type equivalence
- Compiling, including lexical, syntactic and static semantic analysis and code generation
- Run Time settings, including storage allocation and structures to support procedures and functions
- Concepts and techniques for the description and implementation of object-oriented and function-oriented languages.

Skills:
Students should acquire the following skills:
• Be able to explain the relevant techniques and concepts in language design and compiler construction using the terminology and notation for the description and implementation of programming languages
• Be able to explain how implementation techniques influence language design
• Be able to reason about concepts and techniques relevant for computer science

Competences:
Students should be able to describe, analyze and implement programming languages and be able to explain each step and the relationship between the phases of a compiler

Teaching form: Course module
Exam form: Oral or written exam
Assessment: External assessment, 7-point scale.
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Principles of Operation Systems and Concurrency (Principper for styresystemer og parallelitet)
5 ECTS (Course)

Objective
Knowledge:
Students should acquire knowledge about the following theories and methods:
• processes and threads: application, realization, state models, multi-threading programming with threads, creation of process or thread and scheduling
• filesystems: namespace, realization of file systems, strategies for space allocation
• memory management, allocation of primary storage: solid subdivision, virtual memory, paged memory, page replacement algorithms, shared storage, copy-on-write, demand paging, frame allocation
• operating system kernel: interrupts, realization of system calls, drivers for peripheral devices, I/O planning and execution, hardware support
• concurrency / parallelism: relative time synchronization, race conditions, mutex, semaphores, monitors, fairness, deadlocks, necessary and sufficient conditions for deadlock, strategies for deadlock handling, multi-core architectures, parallel programming, techniques and tools for simultaneous-/parallel programming, inter-thread-/process communication

Skills:
Students should acquire the following skills:
be able to explain the establishment, structure, functionality and operation of control systems accurately and using the correct terminology and notation

be able to analyze simple programs which make use of parallelism and / or simultaneity

be able to apply relevant techniques to ensure mutual exclusion, fairness and absence of deadlock in simple concurrent / parallel systems

One or more additional advanced topics may be included in the course, e.g. further principles for realization of parallelism or techniques applicable to operating systems.

Competences:
Using the synthesis of the concepts and techniques, students should be able to develop system close simple programs that take advantage of parallelism and / or concurrency.
The student must be able to acquire new knowledge about operating systems and programming of concurrent and parallel systems.

Teaching form: Course module
Exam form: Oral or written exam
Assessment: External, 7-point scale
Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Advanced Topics in Machine Intelligence (Avancerede emner inden for maskinintelligens)

5 ECTS (Course)

Objective: Knowledge:
The student should gain knowledge of advanced topics dealing with methods and application of machine intelligence, e.g.:
• advanced techniques in data mining
• advanced methods for reasoning and decision making under uncertainty
• agent-based design of intelligent systems
• intelligent web-based systems

Skills:
Students should achieve skills to identify and use advanced techniques from machine intelligence for constructing intelligent systems

Competencies:
The student should be able to understand advanced methods for the design of intelligent systems and to analyze their applicability and efficacy in solving specific tasks.
Title: Test and Verification (Test og verifikation)

5 ECTS (course)

Prerequisites: Knowledge and skills in computer architecture, principles of parallelism, concurrency and operating systems, and syntax and semantics.

Objective: Knowledge:
Students should achieve knowledge on the following theories and methods:

Testing:
• classical test techniques, models for formal testing, software tools for automated testing, test specification, test generation and test execution

verification:
• formal models of software systems behavior, software tools for verification

Skills:
The student should be able to:
• explain accurately and using the subject's terminology and notation for properties and behavior of formal models of software systems
• apply relevant techniques to plan and conduct tests

Competences:
The student should by synthesis of the concepts and techniques of the discipline be able to:
• describe key aspects of a software system using formal models
• assess the usefulness of various test techniques in a software system in a given context

Teaching form: Course

Exam form: Oral or written exam

Assessment: Internal assessment, 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations
Title: Advanced Topics in Modeling and Verification
(Avancerede emner inden for modellering og verifikation)

5 ECTS (course)

Objective:

Knowledge:
Students should achieve knowledge of recent research on advanced mathematical models for the formal description and verification of programmes, software systems and programming languages. E.g. Binary Decision Diagrams (BDD), SAT-algorithms, predicate logic, Petri nets, temporal logician and mobile process calculi.

Skills:
The student should achieve the following skills:
• the ability to explain course concepts and important theories precisely using the terminology and notation of the discipline
• apply methods for specification and verification based on formal models
• be able to make use of the necessary writing skills in these contexts

Competencies:
The student should be able to use formal models and associated verification tools for description, analysis and verification of software systems.

Teaching form: Course

Exam form: Oral or written exam

Assessment: Internal assessment according to 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations
3. semester, IT9

Title: Pre-Specialisation in Computer Science (Forspecialisering i datalogi)

20 ECTS (Project)

Prerequisites: The module adds to knowledge obtained in IT7 and IT8 project and course modules. A course module from IT9 is followed simultaneously with project work.

Purpose: The student should gain insight into a current research problem in computer science and be able to communicate this problem so that the student can make the thesis on this basis.

Reason: University educations are research-based educations. On the master programmes, all students must achieve in-depth insight into current research issues and methods.

Objective: Knowledge:
After having completed the project module, the student should be able to:
• demonstrate in-depth knowledge and overview of a current problem within the research area of computer science.

Skills:
After having completed the project module, the student should be able to:
• use and reason about relevant concepts and techniques within the discipline
• use and create theories within the discipline in the formulation and analysis of a problem within the research area
• communicate a current computer science problem as well as the related concepts in the framework of the research area

Competencies:
After having completed the project module, the student should be able to:
• apply concepts and reasoning within the discipline to formulate and analyse a current open challenge within the research area

Teaching form: Project work, including:
• formulation and analysis of a problem in the research area
• reasoned reflection on solving this problem

Exam form: Oral exam based on project report

Assessment: External assessment according to 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations
Title: Entrepreneurship (Entreprenørskab)

5 ECTS (Course)

Objectives: Knowledge:
The student should achieve knowledge about entrepreneurship and business development related to software (information and communication technologies) including typically:
• different scientific approaches to entrepreneurship, including effectuation
• intra-/entrepreneurship
• competition and market conditions
• business models and business plans
• intellectual property rights
• market development and marketing
• growth strategies
• open entrepreneurship

Skills:
The student should achieve the following skills:
• the ability to explain course concepts precisely using the professional terminology of the discipline
• the ability to use those concepts to explain practical and empirical (case based) contexts

Competencies:
The student should be able to formulate, develop and present their own software-related business ideas to a qualified audience.

Teaching form: Course

Exam form: Oral or written exam

Assessment: Internal assessment, pass/fail

Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Specialisation Course in Database Technology (Specialiseringskursus i databaseteknologi)

5 ECTS (Course)

Prerequisites: The module adds to knowledge obtained in 1st and 2nd semester of the MSc programme
Objective:

Knowledge:
The student should achieve in-depth insight into key issues in contemporary research in database technology.

Skills:
Based on a scientific article in the course's central themes, the student should be able to:
• give a clear and understandable presentation of the article's key issues, including its premises, problem(s), theory, methods, results and conclusions
• explain relevant/key theories, methods, and arguments presented in the article

Competences:
Based on a scientific article in the course's central themes, the student should be able to:
• relate the theories, methods, and results presented in the article to the course topics
• assess the proposed solutions, results and/or conclusions of the article as well as assess their qualities and practicality and put them into perspective.

Teaching form:
Course

Exam form:
The student gives a lecture of 30 minutes on a defined scientific subject area (typically in the form of an article) in relation to issues addressed in the course. The selection of subject area and the framing of the task to each student are made by the course lecturer, usually in consultation with the student's project supervisor. The student is given 7 days of preparation. After the lecture, the examiner and censor can ask questions related to the student's presentation of the theme. This does not normally exceed 10 minutes.

Assessment:
External assessment, 7-point-scale

Evaluation criteria:
Are stated in the Joint Programme Regulations

Title:
Specialisation Course in Distributed Systems
(Specialiseringskursus i distribuerede systemer)

5 ECTS (Course)

Prerequisites:
The module adds to knowledge obtained in 1st and 2nd semester of the MSc programme

Objective:
Knowledge:
The student should achieve in-depth insight into key issues in contemporary research in distributed systems.

Skills:
Based on a scientific article in the course's central themes, the student should be able to:
• give a clear and understandable presentation of the article's key issues, including its premises, problem(s), theory, methods, results and conclusions
• explain relevant / key theories, methods, and arguments presented in the article

**Competences:**
Based on a scientific article in the course's central themes, the student should be able to:
• relate the theories, methods, and results presented in the article to the course topics
• assess the proposed solutions, results and/or conclusions of the article as well as assess their qualities and practicality and put them into perspective.

**Teaching form:** Course

**Exam form:** The student gives a lecture of 30 minutes on a defined scientific subject area (typically in the form of an article) in relation to issues addressed in the course. The selection of subject area and the framing of the task to each student are made by the course lecturer, usually in consultation with the student's project supervisor. The student is given 7 days of preparation. After the lecture, the examiner and censor can ask questions related to the student's presentation of the theme. This does not normally exceed 10 minutes.

**Assessment:** External assessment, 7-point-scale

**Evaluation criteria:** Are stated in the Joint Programme Regulations

---

**Title:** Specialisation Course in Human-Computer Interaction (Specialiseringskursus i menneske-maskine interaktion)

5 ECTS (Course)

**Prerequisites:** The module adds to knowledge obtained in 1st and 2nd semester of the MSc programme

**Objective:**

**Knowledge:**
The student should achieve in-depth insight into key issues in contemporary research in Human-Computer interaction.

**Skills:**
Based on a scientific article in the course's central themes, the student should be able to:
• give a clear and understandable presentation of the article's key issues, including its premises, problem(s), theory, methods, results and conclusions
• explain relevant / key theories, methods, and arguments presented in the article

**Competences:**
Based on a scientific article in the course's central themes, the student should be able to:
• relate the theories, methods, and results presented in the article to the course topics
• assess the proposed solutions, results and/or conclusions of the article as well as assess their qualities and practicality and put them into perspective.

Teaching form: Course

Exam form: The student gives a lecture of 30 minutes on a defined scientific subject area (typically in the form of an article) in relation to issues addressed in the course. The selection of subject area and the framing of the task to each student are made by the course lecturer, usually in consultation with the student's project supervisor. The student is given 7 days of preparation. After the lecture, the examiner and censor can ask questions related to the student's presentation of the theme. This does not normally exceed 10 minutes.

Assessment: External assessment, 7-point-scale

Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Specialisation Course in Semantic and Verification (Specialiseringskursus i semantic og verifikation)

5 ECTS (Course)

Prerequisites: The module adds to knowledge obtained in 1st and 2nd semester of the MSc programme

Objective: Knowledge: The student should achieve in-depth insight into key issues in contemporary research in mathematical modelss for formal description and verification of programmes, software systems and programming languages.

Skills: Based on a scientific article in the course's central themes, the student should be able to:
• give a clear and understandable presentation of the article's key issues, including its premises, problem(s), theory, methods, results and conclusions
• explain relevant / key theories, methods, and arguments presented in the article

Competences: Based on a scientific article in the course's central themes, the student should be able to:
• relate the theories, methods, and results presented in the article to the course topics
• assess the proposed solutions, results and/or conclusions of the article as well as assess their qualities and practicality and put them into perspective.

Teaching form: Course

Exam form: The student gives a lecture of 30 minutes on a defined scientific subject area (typically in the form of an article) in relation to issues addressed in the course. The selection of subject area and the framing of the task to each student are
made by the course lecturer, usually in consultation with the student's project supervisor. The student is given 7 days of preparation. After the lecture, the examiner and censor can ask questions related to the student's presentation of the theme. This does not normally exceed 10 minutes.

Assessment: External assessment, 7-point-scale

Evaluation criteria: Are stated in the Joint Programme Regulations

Title: Specialisation Course in Machine Intelligence (Specialiseringskursus i maskinintelligens)

5 ECTS (Course)

Prerequisites: The module adds to knowledge obtained in 1st and 2nd semester of the MSc programme

Objective: Knowledge:
The student should achieve in-depth insight into key issues in contemporary research in machine intelligence, such as datamining and machine learning, graphical models, and intelligent web systems.

Skills:
Based on a scientific article in the course's central themes, the student should be able to:
• give a clear and understandable presentation of the article's key issues, including its premises, problem(s), theory, methods, results, and conclusions
• explain relevant / key theories, methods, and arguments presented in the article

Competences:
Based on a scientific article in the course's central themes, the student should be able to:
• relate the theories, methods, and results presented in the article to the course topics
• assess the proposed solutions, results and/or conclusions of the article as well as assess their qualities and practicality and put them into perspective.

Teaching form: Course

Exam form: The student gives a lecture of 30 minutes on a defined scientific subject area (typically in the form of an article) in relation to issues addressed in the course. The selection of subject area and the framing of the task to each student are made by the course lecturer, usually in consultation with the student's project supervisor. The student is given 7 days of preparation. After the lecture, the examiner and censor can ask questions related to the student's presentation of the theme. This does not normally exceed 10 minutes.

Assessment: External assessment, 7-point-scale
Title: Specialisation Course in Systems Development (Specialiseringskursus i systemudvikling)

5 ECTS (Course)

Prerequisites: The module adds to knowledge obtained in 1st and 2nd semester of the MSc programme

Objective: Knowledge: The student should achieve in-depth insight into key issues in contemporary research in systems development.

Skills: Based on a scientific article in the course's central themes, the student should be able to:
• give a clear and understandable presentation of the article's key issues, including its premises, problem(s), theory, methods, results and conclusions
• explain relevant / key theories, methods, and arguments presented in the article

Competences: Based on a scientific article in the course's central themes, the student should be able to:
• relate the theories, methods, and results presented in the article to the course topics
• assess the proposed solutions, results and/or conclusions of the article as well as assess their qualities and practicality and put them into perspective.

Teaching form: Course

Exam form: The student gives a lecture of 30 minutes on a defined scientific subject area (typically in the form of an article) in relation to issues addressed in the course. The selection of subject area and the framing of the task to each student are made by the course lecturer, usually in consultation with the student's project supervisor. The student is given 7 days of preparation. After the lecture, the examiner and censor can ask questions related to the student's presentation of the theme. This does not normally exceed 10 minutes.

Assessment: External assessment, 7-point-scale

Evaluation criteria: Are stated in the Joint Programme Regulations
Title: Specialisation Course in Programming Technology (Specialiseringskursus i programmeringsteknologi)

5 ECTS (Course)

Prerequisites: The module adds to knowledge obtained in 1st and 2nd semester of the MSc programme

Objective: Knowledge:
The student should achieve in-depth insight into key issues in contemporary research in programming technology.

Skills:
Based on a scientific article in the course's central themes, the student should be able to:
• give a clear and understandable presentation of the article's key issues, including its premises, problem(s), theory, methods, results and conclusions
• explain relevant / key theories, methods, and arguments presented in the article

Competences:
Based on a scientific article in the course's central themes, the student should be able to:
• relate the theories, methods, and results presented in the article to the course topics
• assess the proposed solutions, results and/or conclusions of the article as well as assess their qualities and practicality and put them into perspective.

Teaching form: Course

Exam form: The student gives a lecture of 30 minutes on a defined scientific subject area (typically in the form of an article) in relation to issues addressed in the course. The selection of subject area and the framing of the task to each student are made by the course lecturer, usually in consultation with the student's project supervisor. The student is given 7 days of preparation. After the lecture, the examiner and censor can ask questions related to the student's presentation of the theme. This does not normally exceed 10 minutes.

Assessment: External assessment, 7-point-scale

Evaluation criteria: Are stated in the Joint Programme Regulations
Title: Master’s Thesis (Kandidatspeciale)

Prerequisites: The module adds to knowledge obtained in project and course modules at IT7-IT9

Purpose: That students are able to formulate, analyse and help solve a current research problem in computer science in an independent, systematic and critical manner through the use of scientific theory and methodology.

Reason: University educations are research-based educations. On the master programmes, all students must achieve in-depth insight into current research issues and methods in a way that this insight can be brought to use in solving research problems.

Objective: Knowledge: After having completed the master’s thesis, the student should be able to:
• demonstrate in-depth knowledge and overview of a current problem within the research area of computer science.

Skills: After having completed the master’s thesis, the student should be able to:
• use and reason about relevant concepts and techniques within the discipline
• use and create theories within the discipline in the formulation and analysis of a problem within the research area
• communicate a current computer science problem as well as the related concepts in the research area framework

Competences: After having completed the master’s thesis, the student should be able to:
• apply concepts and reasoning within the discipline to describe and analyse a current problem within the research area

Teaching form: Project work, including formulation, analysis and contribution to the resolution of a current research problem within computer science and usually follows the subject of the project module on the third semester (IT9).

Exam form: Oral exam based on project report

Assessment: External assessment according to 7-point scale

Evaluation criteria: Are stated in the Joint Programme Regulations
Chapter 4: Entry into Force, Interim Provisions and Revision

The curriculum is approved by the Dean of The Technical Faculty of IT and Design and enters into force as of September 2017.

Chapter 5: Other Provisions

5.1 Rules concerning written work, including the Master’s thesis

In the assessment of all written work, regardless of the language it is written in, weight is also given to the student's spelling and formulation ability, in addition to the academic content. Orthographic and grammatical correctness as well as stylistic proficiency are taken as a basis for the evaluation of language performance. Language performance must always be included as an independent dimension of the total evaluation. However, no examination can be assessed as ‘Pass’ on the basis of good language performance alone; similarly, an examination normally cannot be assessed as ‘Fail’ on the basis of poor language performance alone.

The Board of Studies can grant exemption from this in special cases (e.g., dyslexia or a native language other than Danish).

The Master’s thesis must include an English summary.\(^1\) If the project is written in English, the summary must be in Danish.\(^2\) The summary must be at least 1 page and not more than 2 pages. The summary is included in the evaluation of the project as a whole.

5.2 Rules concerning credit transfer (merit), including the possibility for choice of modules that are part of another programme at a university in Denmark or abroad

In the individual case, the Board of Studies can approve successfully completed (passed) programme elements from other Master’s programmes in lieu of programme elements in this programme (credit transfer). The Board of Studies can also approve successfully completed (passed) programme elements from another Danish programme or a programme outside of Denmark at the same level in lieu of programme elements within this curriculum. Decisions on credit transfer are made by the Board of Studies based on an academic assessment. See the Joint Programme Regulations for the rules on credit transfer.

5.3 Rules for examinations

The rules for examinations are stated in the Examination Policies and Procedures published by The Technical Faculty of IT and Design on their website.

5.4 Exemption

In exceptional circumstances, the Board of Studies study can grant exemption from those parts of the curriculum that are not stipulated by law or ministerial order. Exemption regarding an examination applies to the immediate examination.

---

1 Or another foreign language (upon approval from the Board of Studies).
2 The Board of Studies can grant exemption from this.
5.5 Rules and requirements for the reading of texts
At programmes taught in Danish, it is assumed that the student can read academic texts in modern Danish, Norwegian, Swedish and English and use reference works, etc., in other European languages. At programmes taught in English, it is assumed that the student can read academic text and use reference works, etc., in English.

5.6 Additional information
The current version of the curriculum is published on the Board of Studies’ website, including more detailed information about the programme, including exams.