Curriculum for the Master of Science Programme in Embedded Software Systems

Aalborg University
September 2015
Preface:
Pursuant to the Danish (Consolidation) Act no. 960 of August 14, 2014 on Universities (the University Act) with subsequent changes, the following curriculum for the Master’s programme in Embedded Software Systems is stipulated. The programme also follows the Framework Provisions and the Examination Policies and Procedures for the Faculty of Engineering and Science.

AAU, October 2014
Lone Leth Thomsen
Chairman of Study Board for Computer Science

Table of contents

Chapter 1: Legal Basis of the Curriculum, etc. ................................................................. 2
  1.1 Basis in ministerial orders ....................................................................................... 2
  1.2 Faculty affiliation .................................................................................................. 2
  1.3 Board of Studies affiliation .................................................................................. 2
  1.4 Body of external examiners .................................................................................. 2

Chapter 2: Admission, Degree Designation, Programme Duration and Competence Profile ................................................................. 2
  2.1 Admission ............................................................................................................. 2
  2.2 Degree designation in Danish and English ............................................................ 2
  2.3 The programme’s specification in ECTS credits .................................................... 2
  2.4 Competence profile on the diploma ..................................................................... 2
  2.5 Competence profile of the programme: ................................................................. 3

Chapter 3: Content and Organization of the Programme ................................................................................................. 3
  1st semester, ESS7 ..................................................................................................... 4
  2nd semester, ESS8 ................................................................................................... 8
  3rd semester, ESS9 ................................................................................................... 14
  4th semester, ESS10 .................................................................................................. 16

Chapter 4: Entry into Force, Interim Provisions and Revision ................................................................................................. 18

Chapter 5: Other Provisions .......................................................................................... 18
  5.1 Rules concerning written work, including the Master’s thesis .............................. 18
  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 ........................................................................................................... 18
  5.3 Rules for examinations ........................................................................................ 18
  5.4 Exemption .......................................................................................................... 18
  5.5 Additional information ........................................................................................ 19

Completion of the Master’s programme ........................................................................ 19
Chapter 1: Legal Basis of the Curriculum, etc.

1.1 Basis in ministerial orders
The Master’s programme in Embedded Software Systems is organized in accordance with the Ministry of Higher Education and Science’s Ministerial Order no. 1520 of December 16, 2013 on Bachelor’s and Master’s Programmes at Universities (the Ministerial Order of the Study Programmes) and Ministerial Order no. 670 of June 19, 2014 on University Examinations (the Examination Order) with subsequent changes. Further reference is made to Ministerial Order no. 1488 of December 16, 2013 (the Admission Order) and Ministerial Order no. 250 of March 15, 2007 (the Grading Scale Order) with subsequent changes.

1.2 Faculty affiliation
The Master’s programme falls under the Faculty of Engineering and Science, 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
Body of external examiners for Computer Science

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

2.1 Admission
Admission to the Master’s programme in Embedded Software Systems requires a Bachelor’s degree in computer science or software engineering.

Students with a Bachelor’s degree in other subjects may be admitted after a specific academic assessment after application to the Board of Studies. 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.polyt. (candidatus/candidata polytechnices) i indlejrede softwaresystemer. The English designation is: Master of Science (MSc) in Engineering (Embedded Software Systems).

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 embedded software engineering 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 software engineering and identify solutions to scientific and engineering problems

Skills
- are proficient in scientific methods and tools and general skills related to development of embedded software
- can evaluate and select among the theories, methods, tools and general skills and, on a sound engineering basis, advance new analyses and solutions
- can communicate research-based knowledge and discuss professional 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 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 (ESS7)</td>
<td>Practical Development of Embedded Systems</td>
<td>15</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td><strong>Three of the following:</strong></td>
<td></td>
<td></td>
<td></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>Real Time Systems</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</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 Distributed Systems</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td>2 (ESS8)</td>
<td>Embedded Software Platforms</td>
<td>15</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td>Test and Verification</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td><strong>Two of the following:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semantics and Verification</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Reconfigurable and Low Energy Systems</td>
<td>5</td>
<td>Pass/Fail</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 Topics in Systems Development</td>
<td>5</td>
<td>7-point scale</td>
<td>Internal</td>
</tr>
<tr>
<td>3 (ESS9)</td>
<td>Embedded Software in the Systems Environment</td>
<td>20</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td>Specialisation Course in Distributed Systems</td>
<td>5</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td>Entrepreneurship</td>
<td>5</td>
<td>Pass/Fail</td>
<td>Internal</td>
</tr>
<tr>
<td>4 (ESS10)</td>
<td>Master's Thesis</td>
<td>30</td>
<td>7-point scale</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td>120</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**1st semester, ESS7**

**Title:** Practical Development of Embedded Systems  
(Praktisk udvikling af indlejrede systemer)

**Scope:** 15 ECTS (Project)

**Prerequisites:** ESS7 course modules should be followed in parallel.

**Objective:**

**Knowledge**
Students who complete the module:
- have knowledge about architectural principles for embedded software
- know standard scheduling techniques
- know how to check real-time constraints for simple, well-structured embedded software
- know basic principles of testing for real-time systems

**Skills**
Students who complete the module:
- Are able to design, implement and test small embedded systems, for instance using Lego Nxt (R) components for prototyping the physical environment.

**Competences**

Students who complete the module:
- can identify a problem within computer science research or applications
- can contribute to a solution of the problem through modelling based on computer science theories
- can analyse and evaluate the contribution
- can analyse and evaluate applications of relevant models for solving the problem

**Type of instruction:** Project

As an integrated part of the project work, the student must participate in the problem-based learning and project management workshop (1 ECTS). Approved participation is required to register for the project exam. See [enclosure 1](#).

**Exam format:** Oral exam

**Assessment:** Internal assessment according to 7-point scale

**Evaluation criteria:** Are stated in the Framework Provisions

---

**Title:** Programming Paradigms  
* (Programmeringsparadigmer)

**Scope:** 5 ECTS (course, elective)

**Prerequisites:** Knowledge of Imperative Programming, Object Oriented Programming, Languages and Compilers, Syntax and Semantics

**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: RealTime Systems (Tidstro software)
Scope: 5 ECTS (course, elective)
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 real-time programming, hardware abstraction, and system near programming, synchronization, atomicity, deadlocks, error handling, communication

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: Permitted aids and exam form determined by the course lecturer
Assessment: Internal assessment, 7-point scale
<table>
<thead>
<tr>
<th>Title:</th>
<th><strong>Computability and Complexity (Beregnelighed og kompleksitet)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope:</td>
<td>5 ECTS (Course, elective)</td>
</tr>
<tr>
<td>Prerequisites:</td>
<td>Knowledge of Discrete Mathematics, Algorithms and data structures, Syntax and Semantics.</td>
</tr>
<tr>
<td>Objective:</td>
<td>Knowledge: Students should achieve knowledge on the following theories and methods:</td>
</tr>
<tr>
<td></td>
<td>Computability:</td>
</tr>
<tr>
<td></td>
<td>• deterministic and nondeterministic Turing machines; decidable and recognizable languages and their properties: Church-Turing thesis</td>
</tr>
<tr>
<td></td>
<td>• acceptance problem for Turing machines; other undecidable problems for Turing machines; reductions and their properties</td>
</tr>
<tr>
<td></td>
<td>Complexity theory:</td>
</tr>
<tr>
<td></td>
<td>• 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</td>
</tr>
<tr>
<td></td>
<td>• space complexity of deterministic and nondeterministic Turing machines; space complexity classes, the relationship between time and space complexity</td>
</tr>
<tr>
<td></td>
<td>Skills:</td>
</tr>
<tr>
<td></td>
<td>Students should achieve the following skills:</td>
</tr>
<tr>
<td></td>
<td>• 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</td>
</tr>
<tr>
<td></td>
<td>• the ability to make use of the necessary writing skills in these contexts</td>
</tr>
<tr>
<td></td>
<td>Competencies:</td>
</tr>
<tr>
<td></td>
<td>The student should be able to apply concepts and techniques from the theory of computability and computational complexity for the analysis of computational problems</td>
</tr>
<tr>
<td>Teaching form:</td>
<td>Course</td>
</tr>
<tr>
<td>Exam form:</td>
<td>Oral or written exam</td>
</tr>
<tr>
<td>Assessment:</td>
<td>External assessment, 7-point scale</td>
</tr>
</tbody>
</table>

**Title:** **Advanced Topics in Distributed Systems (Avancerede emner indenfor distribuerede systemer)**
Scope: 5 ECTS (course, elective)

Prerequisites: Knowledge of computer architectures and concurrent systems, programming skills, system programming. Or prerequisites corresponding to a qualifying undergraduate education, and in some cases, the relevant parts of the learning objectives of courses offered at the qualifying bachelor. These prerequisites can be obtained at the beginning of the course through special activities integrated into the course.

Objective: Knowledge:
The student will gain knowledge of advanced theories and methods in distributed and embedded systems:
- advanced infrastructures and applications; for example, grid, cloud, peer-to-peer, or parallel / multi-core systems
- system and network programs for embedded systems
- examples of distributed embedded systems, such as ad-hoc sensor networks, home automation
- distributed algorithms such as algorithms for reciprocal 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

2nd semester, ESS8

Title: Embedded Software Platforms (Platforme for indlejret software)
Scope: 15 ECTS (Project)
Prerequisites: ESS8 course modules should be followed in parallel.
Objective: Knowledge:
Students who complete the module:

- have knowledge about common hardware platforms for embedded systems software
- know standard device interface principles
- know application areas for co-processing techniques
- have knowledge about key principles for hardware/software co-design
- know standard scheduling techniques
- know how to check real-time constraints for simple, well-structured embedded software
- know basic principles of testing for real-time systems
- are familiar with programming techniques for multicores and multiprocessor platforms
- have thorough knowledge of networks for real-time systems

Skills:
Students who complete the module:

- are able to design, implement and test embedded software for unconventional platforms which for instance include wireless sensor networks, FPGA-based (co)processors, or heterogeneous multiprocessors.

Competences
Students who complete the module:

- can identify a problem within computer science research or applications
- can contribute to a solution of the problem through modelling based on computer science theories
- can analyse and evaluate the contribution
- can analyse and evaluate applications of relevant models for solving the problem

Type of instruction: Project
Exam format: Oral exam,
Assessment: External assessment, 7-point scale
Evaluation criteria: are stated in the Framework Provisions

---

**Title:** Test and Verification (Test og verifikation)

**Scope:** 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 Framework Provisions

**Title:** Semantics and Verification (Semantik og verifikation)

**Scope:** 5 ECTS (course)

**Prerequisites:** Knowledge of Syntax and Semantics, Computability and Complexity.

**Objective:**
**Knowledge:** The student should acquire knowledge of advanced mathematical models for the formal description and verification of programs, software systems and programming languages, focusing on parallel and communicating systems. Specifically, students should gain knowledge about:
• transition systems
• process algebra, e.g. CCS
• bisimulation
• Hennessy-Milner logic with recursion
• Tarski's theorem on fixed points
• models and reasoning methods for real-time systems, such as Timed CCS and time machines
• probabilistic modeling and reasoning techniques, such as probabilistic process calculi, equivalences and logics
• verification techniques for real-time models
• any other verification

Course module can also include other formal models.

**Skills:**
The student should achieve the following skills:
• ability to explain accurately and using subject terminology and notation for important theories for description and analysis of reactive systems
• ability to use verification tools based on formal models
• ability to make use of the necessary writing skills in these contexts
Competences:
The student should be able to use formal models and associated verification tools for the verification of software systems.

Teaching form: Course
Exam form: Oral or written exam
Assessment: Internal assessment, 7-point scale

Title: Reconfigurable and Low Energy Systems (Rekonfigurerbare og energi-minimale systemer)

Scope: 5 ECTS (course)
Prerequisites: Fundamental hardware/software components for embedded systems, programming languages and design methodologies.

Objective: Knowledge:
Students who complete the module:
Must have knowledge about;
• computational complexity theory
• cost functions and models of computation
• graph representation and -analysis of Digital Signal Processing (DSP) algorithms
• interaction between DSP algorithms and real-time Hardware/Software (HW/SW) architectures
• design and optimization of dedicated HW architectures/co-processors
• scheduling methodologies for multiple processors/functional units
• hardware/software metric-estimation and partitioning
• static and dynamic reconfiguration management
• technologies and methods for partial reconfiguration
• models for power consumption in digital circuits
• power/energy optimization techniques in HW/SW architectures
• essential issues of probabilistic embedded computing
• battery models and management
• energy harvesting methodologies

Skills:
Students who complete the module:
• Must be able understand advanced terms, concepts, and methods, their application as well
• as limitations in the context of time-, area-, or energy optimal/constrained mapping of DSP
• algorithms onto real-time HW/SW architectures.

Competencies:
Students who complete the module:
• The student must be able to apply the proper terminology in oral and written communication
• and documentation within the scientific domains of DSP algorithms, and application specific
• HW/SW architectures.

Teaching form: Course
Exam form: Oral or written exam
Assessment: Internal assessment, pass/fail

Title: Mobile Software Technology (Mobil softwareteknologi)
Scope: 5 ECTS (course, elective)
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 locations 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

Title: Advanced Topics in Systems Development (Avancerede emner inden for systemudvikling)

Scope: 5 ECTS (course, elective)
Prerequisites: Bachelor's degree in Computer Science or Software or equivalent

Objective: Knowledge: The student should gain knowledge about advanced topics in systems engineering in theory and practice. The topics may include, but are not limited to:
• analysis of systems engineering practices
• systems development methodologies, processes and - skills
• organization and management of systems development
• developing systems for complex structures, e.g. supporting cooperation in organizations, knowledge-intensive systems and information infrastructure

Skills: Within the course subject matter, the student should:
• be able to understand and present the course topics including premises, problems, theories, methods, results and conclusions
• be able to apply theories and methods to analyze and describe a problem in practical systems development
• demonstrate a critical approach to theories and methods in systems development

Competencies: The student should able to describe, analyze and evaluate a specific practice of a systems development company including:
• relate to the theories and empirical methods of the discipline
• demonstrate perspective in relation to selected topics such as: requirements management, quality management, outsourcing, distributed development, agile processes, and model-driven processes

Teaching form: Course
Exam form: Oral or written exam
Assessment: Internal assessment according to 7-point scale
3rd semester, ESS9

Title: Embedded Software in the Systems Environment (Indlejret software i systemsammenhæng)

Scope: 20 ECTS (Project)

Prerequisites: ESS7 and ESS8 project and course modules and that a course module from ESS9 is followed simultaneously with project work

Purpose: The student should gain insight into a current research problem within embedded software systems and be able to communicate this problem so that the student can make his/her 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 embedded software systems.

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 research area framework

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

Teaching form: Project work, including:
• formulation and analysis of a problem in a current issue 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


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

Scope: 5 ECTS (course)
Prerequisites: 1st and 2nd semester of the MSc education or the like

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 compensation issues, including its premises, issue(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: Entrepreneurship (Entreprenørskab)

Scope: 5 ECTS (course)

Prerequisites: BSc in Computer Science, Software or the like

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

4th semester, ESS10
Title: Master's Thesis
(Kandidatspeciale)
Scope: 30 ECTS (Project)
Prerequisites: Project and course modules at ESS7-ESS9
Purpose: That students are able to formulate, analyze and help solve a current research
problem in machine intelligence 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 embedded software systems.

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 formulate and analyze a current issue within the research area

Teaching form: Project work, including formulation, analysis and contribution to the resolution of a current research problem within machine intelligence and usually follows the subject of the project module on the third semester (DE9)

Exam form: Oral exam based on project report

Assessment: External assessment according to 7-point scale

Chapter 4: Entry into Force, Interim Provisions and Revision

The curriculum is approved by the Dean of the Faculty of Engineering and Science and enters into force as of September 2015.

Students who wish to complete their studies under the previous curriculum must conclude their education by the summer examination period 2016 at the latest, since examinations under the previous curriculum are not offered after this time.

In accordance with the Framework Provisions for the Faculty of Engineering and Science at Aalborg University, the curriculum must be revised no later than 5 years after its entry into force.

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. If the project is written in English, the summary must be in Danish. 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 Framework Provisions 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 Faculty of Engineering and Science 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 with approval by the Board of Studies.
2 The Board of Studies can grant exemption from this.
5.5 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.

Completion of the Master's programme
The Master's programme must be completed no later than four years after it was begun.
<table>
<thead>
<tr>
<th>Enclosure 1:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title:</strong> Problem based learning and project management (Problembaseret læring og projektledelse)</td>
</tr>
<tr>
<td><strong>Size:</strong> 1 ECTS</td>
</tr>
<tr>
<td><strong>Prerequisites:</strong> None</td>
</tr>
<tr>
<td><strong>Objectives:</strong> The objective is to make newly started Master students coming from institutions other than AAU prepared to enter the problem based learning environment at AAU and manage study projects in close collaboration with peers. After completion of the course the student should have acquired:</td>
</tr>
<tr>
<td>- <strong>Knowledge</strong> about AAU as a frame of study and student life in Aalborg</td>
</tr>
<tr>
<td>- <strong>Knowledge</strong> 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</td>
</tr>
<tr>
<td>- <strong>Knowledge</strong> to identify similarities and differences between the Aalborg PBL study environment and previous study environments, incl. strengths and weaknesses in both environments</td>
</tr>
<tr>
<td>- <strong>Skills</strong> to structure project management activities based on a well-formulated problem formulation</td>
</tr>
<tr>
<td>- <strong>Skills</strong> to assess project documentation based on scientific codes of conduct.</td>
</tr>
<tr>
<td>- <strong>Competences</strong> to plan for effective collaborative learning in an intercultural environment and manage group conflicts</td>
</tr>
<tr>
<td>- <strong>Competence</strong> to reflect on, plan and manage a study project in a PBL learning environment</td>
</tr>
<tr>
<td><strong>Type of instruction:</strong> Three half day workshops</td>
</tr>
<tr>
<td><strong>Exam format:</strong> The assessment is performed based on active participation in the arranged workshops.</td>
</tr>
<tr>
<td><strong>Evaluation criteria:</strong> The criteria for the evaluation are specified in the Framework Provisions.</td>
</tr>
</tbody>
</table>