Linnæus University

Jnr: 2018/3464-3.1.1.3

Programme syllabus

Faculty of Technology

Programvaruteknik, masterprogram, 120 högskolepoäng
Software Technology, Master Programme, 120 credits

Level
Second Level

Date of Ratification
Approved 2009-09-15
Revised 2018-12-07 by the Faculty Board within the Faculty of Technology
The programme syllabus is valid from autumn semester 2019

Prerequisites
General entry requirements for second-cycle studies and specific entry requirements:

- Bachelor Degree in Computer Science or the equivalent, and a solid practical experience of object-oriented programming.
- English B/6 or the equivalent.

Description of Programme

Software can be considered the invisible infrastructure of the digitized economy, and is everywhere, from stand-alone products to an increasingly important part of traditional products. There is, therefore, a need for well-trained staff to develop the software that governs today's and tomorrow's systems.

The degree programme advances the students' prior knowledge of computer science and software development. The studies prepare to work in organizations where software is used and developed, and a new graduate from the program is expected to after some time embrace all roles related to software development in a software development project, from technical expert to project manager. The degree program also provides a good foundation for starting and running a business as well as an academic career in computer science and education.

Objectives

Central degree objectives in accordance with the Higher Education Ordinance

Knowledge and understanding

For a Degree of Master (120 credits) the student shall

- demonstrate knowledge and understanding in the main field of study, including both broad knowledge of the field and a considerable degree of specialised knowledge in certain areas of the field as well as insight into current research and

development work, and
- demonstrate specialised methodological knowledge in the main field of study.

Competence and skills

For a Degree of Master (120 credits) the student shall

- demonstrate the ability to critically and systematically integrate knowledge and analyse, assess and deal with complex phenomena, issues and situations even with limited information,
- demonstrate the ability to identify and formulate issues critically, autonomously and creatively as well as to plan and, using appropriate methods, undertake advanced tasks within predetermined time frames and so contribute to the formation of knowledge as well as the ability to evaluate this work,
- demonstrate the ability in speech and writing both nationally and internationally to clearly report and discuss his or her conclusions and the knowledge and arguments on which they are based in dialogue with different audiences, and
- demonstrate the skills required for participation in research and development work or autonomous employment in some other qualified capacity.

Judgement and approach

For a Degree of Master (120 credits) the student shall

- demonstrate the ability to make assessments in the main field of study informed by relevant disciplinary, social and ethical issues and also to demonstrate awareness of ethical aspects of research and development work,
- demonstrate insight into the possibilities and limitations of research, its role in society and the responsibility of the individual for how it is used, and
- demonstrate the ability to identify the personal need for further knowledge and take responsibility for his or her ongoing learning.

Program Specific Objectives

Competence and skills

For a Degree of Master (120 credits) the student shall

- demonstrate the ability to create, analyze, and critically evaluate different software technological solutions,
- demonstrate the ability to model, simulate, predict, and assess events even with limited information,
- demonstrate the ability to design and manage software systems while considering human conditions and needs, and society's objectives for economically, socially and ecologically sustainable development, and
- demonstrate an ability for teamwork and collaboration in groups with different compositions.

Judgement and approach

For a Degree of Master (120 credits) the student shall

- demonstrate the ability to seek additional knowledge and to develop their skills continuously, and
- demonstrate an understanding of the capabilities and limitations of software, its role in society and people's responsibility for its use, including social and economic aspects, environmental and safety issues.

Content

Programme Overview

The main area of education is computer science. A program manager has overall responsibility for the program. A program council is responsible for the program's development and quality.

The program consists of a total of 120 higher education credits, which corresponds to two years full-time studies.

In addition to in-depth knowledge of computer science with a focus on model-driven development, data-intensive systems, and visualization and data analysis, the degree programme also provides good knowledge in problem-solving. The degree programme is project-driven and contains three larger projects, where, e.g., communication, professional skills and approaches, and system thinking are practiced. These projects cover the entire development cycle, from needs and idea to operation, in a realistic environment.

Programme Courses

The listed points below (in hp) refer to higher education credits. More information about the courses is available in the respective course syllabus.

Year 1

Systems Modeling and Simulation, 5 hp, A1N*: Advanced Course in model-based development that addresses how systems with high demands on, e.g., security and reliability can be modeled and simulated to verify properties before they are implemented.

Code Transformation and Interpretation, 5 hp, A1N*: Advanced Course in model-based development that addresses how computer languages can be formulated and how they can be translated, e.g., from Java to executable code. It introduces many essential principles such as grammar, type inference, semantic analysis, and advanced state machines.

Project in Model-based Development, 10 hp, A1N*: A project course in model-based development where students apply knowledge in modeling, architecture, and simulation to carry out an agile development project in a realistic environment and with an open problem. Particular emphasis is placed on how to apply agile methods.

Formal methods, 5 hp, A1N* (can be replaced by an elective course): Advanced Course in model-based development that addresses how properties of models and software programs can be formally verified, e.g., security.

Selected topics in computer science, 5 hp, A1N* (can be replaced by an elective course): Independent studies course.

Machine Learning, 5 hp, A1N*: Advanced course in data-intensive systems that addresses artificial intelligence and learning systems, with a focus on statistical machine learning and clustering.

Parallel Computing, 5 hp, A1N*: Advanced course in data-intensive systems that addresses how parallel computer systems and accelerators can be used to solve problems that deal with large amounts of data. Common architectures and methods for decomposing problems are discussed, with a focus on graphics processors and solutions to linear algebra problems.

Project in Data Intensive Systems, 10 hp, A1N*: A project course in data-intensive systems where students apply knowledge in machine learning and parallel programming to carry out an agile development project in a realistic environment and with an open problem. Particular emphasis is placed on effective agile methods, lean agile.

Deep Machine Learning, 5 hp, A1N*: (can be replaced by an elective course). Advanced course in data-intensive systems that addresses deep (machine) learning and artificial neuron networks.

Lean Startup, 5 hp, G2F (can be replaced by an elective course): Course in economics that deals with entrepreneurship and innovation, as well as how to start, drive, and change businesses. The course takes a practical approach and builds on previous experiences from lean and agile.

Year 2

Information Visualization, 5 hp, A1N*: Advanced course in visualization and data analysis that addresses how visualizations can help people analyze and understand abstract data. Interaction and user experience are also addressed.

Data Mining, 5 hp, A1F*: Advanced course in visualization and data analysis that addresses methods for creating meaning in unstructured data, such as methods for analysis of (social) networks.

Project In Visualization and Data Analysis, 10 hp, A1F, A project course in visualization and data analysis where students apply knowledge in visualization and data extraction to carry out an agile development project in a realistic environment and with an open problem. Particular emphasis is placed on independently carrying out an agile project.

Advanced Information Visualization and Applications, 5 hp, A1F* (can be replaced by an elective course): Advanced course in visualization and data analysis that addresses how information visualization can be used for a range of applications in bioinformatics, geography and software development.

Scientific Methods in Computer Science, 5 hp, A1N*: Advanced course in scientific methods that addresses current research and methods in computer science. The course is a seminar course where the students present and discuss published scientific results.

Independent Project in Computer Science, 30 hp, A2E*: An independent project where the student will develop in-depth knowledge, understanding, abilities, and approaches in the context of the degree programme. The thesis involves a comprehensive understanding of previously acquired knowledge.

* Courses in the main area Computer Science.

Societal relevance

Students will encounter representatives of working life on several occasions during the program, e.g., several courses are planned to have guest lectures. The degree programme includes three projects and an independent project that can be carried out in collaboration with industry or other organizations. The courses are designed in cooperation with industry representatives to ensure that they address realistic problems and issues.

Internationalization

The third term can be spent at a foreign university. The university and study plan is decided in together with the program manager.

Scope of Programme

The technology and economy are shifting from hardware to software, and software is an increasingly important part of traditional products, from cars to washing machines. Since software represents a cross-section of society, the degree programme needs to address several perspectives.

A large share of software development happens in large and often international teams, so concepts such as (social) sustainable development, gender, diversity, etc. must be addressed by the degree programme.

As software is becoming an essential part of society, the degree program needs to discuss the role of software and its consequences. What risk do data storage carry and what implications might security and reliability problems have? Which part and what responsibilities will the individual software engineer have in this, and what ethical questions should be considered? There is also a need to consider technical sustainability, i.e., how to develop software systems and standards with a long lifespan.

Since software use is increasing, there is also a need to consider usability, user experience, and accessibility, both in society and the degree programme.

The degree program places considerable emphasis on these perspectives, which are discussed in theoretical courses and practiced in project courses.

Quality Development

For each course, a number of students will be appointed as course representatives to represent the student group during the course evaluation. These students will meet with teachers and/or the course manager at a few occasions during the course. The courses are evaluated through a written questionnaire, and after this has been compiled and analyzed, the program manager, responsible teachers, and course representatives meet and write an evaluation report and an action plan (if needed). The evaluation report and action plan from the previous year's course shall be available via the course room or the course website.

The program is evaluated yearly by the program council based on course evaluations, governing documents, and feedback from industry and alumni. The result of this evaluation will be presented to students and teachers at a seminar during the spring semester. The previous year's evaluation report shall be available from the program room or the program website.

Degree Certificate

After completing their studies which correspond to the requirements stated in the Higher Education Ordinance and in Linnaeus University’s local Degree Ordinance, students may apply for a degree. Students who have completed the Master´s programme in Software Technology can obtain the following degree:

Master of Science (120 credits) with specialisation in Software Technology. Main field of study: Computer Science.

The degree certificate is bilingual (Swedish/English). A Diploma Supplement (English) will be provided along with the degree certificate.

Other Information

The following requirements are set for the student to continue to the next term of the program. To continue to term 2, the student must have completed at least 10 higher education credits with passing grades. To continue to term 3, the student must have completed at least 40 higher education credits, including at least one of the projects, with passing grades. To continue to term 4 and start the independent project, the student must have completed the Planning Documents in Scientific methods in computer science with grade E or higher. Students who do not meet these requirements should contact the program's study counselor and program manager to create an individual study plan.

The degree programme is conducted in such a way that the course participants' experiences and knowledge are made visible and developed. This means, for example, that we have an inclusive approach and strive for no one to feel excluded. This can be expressed in different ways in a course, for example by using gender neutral examples.