Curriculum for the Master of Science Programme in Computer Science at the IT University of Copenhagen

Curriculum of 1 August 2024

Content

Background
Chapter 1 Programme Title and Objectives
Chapter 2 Programme Structure, Content and Programme Language
Chapter 3 General rules and Miscellaneous Regulation
Chapter 4 Date of Commencement and Transitional Regulations
Appendix

Background

This curriculum for the Master of Science Programme in Computer Science, has been drawn up by the Board of Studies at the IT University of Copenhagen (henceforth referred to as the IT University). The curriculum has been drawn up in compliance with the current legislation governing bachelor's and master's (Candidatus) programmes at the universities.

Students enrolled in the above MSc study programmes with study start from autumn of 2024 study according to this curriculum.

Chapter 1

Programme Title and Objectives

Programme Title

Section 1. A student, who has completed the programme, has the right to use the title candidatus/candidata scientiarum (cand.scient.) i datalogi.

Subsection 2. The title in English is Master of Science (MSc) in Computer Science.

Programme Objectives

Section 2. The purpose of this programme is to provide students with the scientific qualifications to identify, formulate, solve and reflect on complex problems relating to computer science.

Subsection 2. The programme prioritises the student's ability to assess, apply and develop the underlying technology and the scientific theories, methods and tools upon which it is based.

Subsection 3. The student must have the ability to independently initiate and carry out collaborative work in professional and multidisciplinary settings. Furthermore, the student must have the ability to engage in global and distributed interaction, drawing on research-based perspectives.

Subsection 4. On the background of the student's preceding bachelor's programme, the programme provides the student with the qualifications to define his or her own academic profile within the field of computer science and to take independent responsibility for his or her own professional development and specialisation.

Subsection 5. Within the framework of the programme, the student can acquire the requisite individual qualifications for specialised posts in business, industry, the public sector, and for research training programmes (PhD programme) in computer science.

Objectives for Learning Output

Section 3 The graduate will develop research-based *knowledge and understanding of, and will be able to reflect on*:

- concepts of programming languages
- analysis of software performance and correctness in theory and practice
- principles of software design and modelling and its applications to real-world scenarios
- Principles of machine learning

Each graduate obtains specialised knowledge and understanding at highest international research level in selected specialisation area.

Subsection 2. The graduate will develop the following research-based *skills*:

- The graduate can master a modern programming platform to implement software
- The graduate can use, assess and develop fundamental processes and practices of software development, such as requirements analysis, architecting, implementing, tuning, validating and documenting software.
- The graduate is able to communicate and discuss the acquired knowledge with both academic peers and non-specialists.

Each graduate obtains specialised skills at highest international research level in selected specialisation area.

Subsection 3. The graduate will develop the following research-based *competences*:

- The graduate can evaluate and customise software tools and technologies that are complex and unpredictable, and design and develop new solutions.
- The graduate can independently initiate and implement collaboration with others in complex and changing contexts.
- The graduate can independently take responsibility for own professional development and specialisation.

Each graduate obtains specialised competences at highest international research level in selected specialisation area.

Chapter 2

Programme Structure, Content and Programme Language

Programme Structure

Section 4. The Master of Science programme requires passes in study activities corresponding to 120 ECTS points comprising a mandatory backbone, specialisation modules, optional modules and a master's thesis.

Subsection 2. The study activities of the programme are composed of modules corresponding to 90 ECTS points and a concluding master's thesis corresponding to 30 ECTS points.

Subsection 3. A visualisation of the programme structure is available at the IT University's online Student handbook.

Programme Content

Section 5. The mandatory backbone of the MSc study programme in Computer Science consists of modules corresponding to 37,5 ECTS points within the first three terms.

The content of the mandatory backbone focuses on advanced techniques and methods for designing, implementing, and maintaining software, drawn from both computer science and software engineering research. In detail, modules contain techniques for designing and implementing efficient and correct software for modern IT systems, including concurrent and parallel software, aspects of machine learning, advanced features of programming languages, and project organisation in heterogeneous working environments.

Subsection 2. The choice of specialisation courses are mandatory. A specialisation comprises modules corresponding to 30 ECTS points within the second and the third term.

Subsection 3. The optional modules comprise 22,5 ECTS points within the second and third term.

Programme Language

Section 7. The MSc Computer Science study programme is conducted in English.

Master Thesis

Section 8. The thesis is worth 30 ECTS points and must document skills in applying scientific theories and methods while working within the study programme's subject area.

Subsection 2. The thesis is placed on the final year of the programme. The student must have obtained 60 ECTS of the programme before writing the thesis.

Subsection 3. The abstract must be written in English or Danish.

Subsection 4. Intended learning outcomes for the Master thesis in Computer Science:

ILO 1 To identify, define, and delimit a relevant research problem within Computer Science.

ILO 2 To identify, justify, and describe relevant means for addressing the research problem. These include academic theories, methods, literature, and tools.

ILO 3 To analyze and relate the selected means to the current state of the art.

ILO 4 To combine the selected means, develop them further if necessary, and apply them in a concerted effort.

ILO 5 To report clearly the achieved results using appropriate scientific terminology and methods.

ILO 6 To discuss the achieved results and relate them to the current state of the art.

Subsection 5. Information on examination for Master theses can be found in the appendix.

Chapter 3

General Rules and Miscellaneous Regulation

Admission Requirements

Section 9. Admission requirements to the MSc in Computer Science:
- A university bachelor's degree or a professional bachelors's degree relating to computer science or software development. It must include a significant amount of programming, foundations and software engineering. The specific requirements may be found in section 10.
- English proficiency corresponding to B-level with a minimum of 3.0 in grade point average.
- Other related degrees, which are comparable to a university or professional bachelor's degree, may fulfil the requirements. ITU will make an individual assessment of the degree in question.

Subsection 2. The following programmes give legal right to admission to Computer Science:

- Software Development (ITU)
- Data Science - Technical Track (ITU) - Only students admitted 2020 and before.

Subsection 3. Programmes that automatically meet the specific requirements:

- Datalogi (AAU)
- Software (AAU)
- Datalogi (AU)
- Datalogi (KU)
- Softwareteknologi (DTU) - Bachelor of Science in Engineering (Software Technology)
- Datalogi (SDU)

Section 10. Specific Requirements

- All requirements 1 to 5
- Two of requirements 6 to 8

Requirements	ECTS
1: Imperative/Object-Oriented/Functional Programming	30
Applicants must have a medium level of ability to program in an imperative/OOP programming language. Additionally, applicants must have been exposed to a mainstream functional programming language. These requirements are usually met by taking an introductory course on programming and working on a medium-sized project. *Course example:* *Introduction to Programming BSGRPRO1KU (15 ECTS) +* *First Year Project: Map of Denmark. Visualization, Navigation, Searching and Route Planning BSFDVNS1KU (15 ECTS) +* *Functional Programming BSFUPRO1KU (7.5 ECTS)*	
2: Discrete Mathematics	5
Applicants must have taken an introductory course on discrete math. The course must include most of the following: propositional and predicate logics, induction principles, basic counting, discrete probability, sets, functions, graphs, trees. *Course example:* *Foundations of Computing: Discrete Mathematics BSFOCDM1KU (7.5 ECTS)*	
3: Algorithms and Data Structures	5
Applicants must have taken an introductory course on algorithms, which includes most of the following: big-O notation, basic analysis of algorithm correctness and complexity, hashing, basic data structures, quicksort, mergesort, DFS, BFS, Dijkstra, Kruskal, Prims (MST), priority queues, binary search trees, red-black trees, tries, union-find. *Course example:* *Algorithms and Data Structures 1408001U (7.5 ECTS)*	
4: Databases	5

Basic concepts of databases, relational databases, basic data analytics. *Course example:* *Introduction to Database Systems BSINDBS1KU (7.5 ECTS)*	
5: Software Engineering	5
Introductory course on software engineering. The course must include most of the following: software processes, requirements, software architecture, testing, version control, development tools. *Course example:* *Software Development and Software Engineering BSSODSE1KU (7.5 ECTS)*	
6: Computer Security	5
Some basic knowledge of computer security. This must include most of the following: adversarial environments, security principles, basic methods for preventing cyberattacks, including cryptography, authentication, authorization. *Course example:* *Security 1 BSSECU11KU (7.5 ECTS)*	
7: Programming Language Design and Implementation	5
The prerequisite includes most of the following: regular expressions, finite state machines, parsing techniques, type checking and inference, garbage collection techniques. *Course example:* *Programs as Data BSPRDAT1KU (7.5 ECTS)*	
8: Operating Systems	5
Some knowledge of how operating systems work. This includes most of the following: some knowledge of a low level programming language such as C, basic system programming concepts (I/Os, network programming, concurrent programming), reflect on the impact of hardware, operating system and compilation process on application programs. *Course example:* *Operating Systems and C BSOPSYC1KU (7.5 ECTS)*	

Section 11. Furthermore, please refer to the IT University's rules and regulation, appendix to this curriculum.

Chapter 4

Date of Commencement and Transitional Regulations

Section 12. This curriculum comes into force 1 September 2024 and applies to all students admitted to the programme from autumn 2024.

Subsection 2. When a new curriculum is published, or in the event of significant changes to this curriculum, transitional regulations will be set out in the curriculum as appendix.

Approved by the Board of Studies ITU 17 December 2024.

Approved by rector Per Bruun Brockhoff   December 2023

Transitional Regulations:

Students admitted before autumn 2024 complete the program under the old curriculum.