Cand. Scient. in Computer Science

M. Carbone, S. Debois, Y. Dittrich, P. Sestoft, A. Wasowski

March 21, 2017 Draft.

Overview

This document describes the design and implementation of the new programme in Computer Science, a two-year education at MSc level built on top of an undergraduate degree (or equivalent).

Cand. scient. in Computer Science is strongly inspired by the Advanced Computing track of the MSc cand. it. in Software Development (SDT). Based on the 2015 external review of the latter and on several recommendations of the employer's panel, ITU has decided to split the SDT programme and form a new programme in Computer Science from the Advanced Computing track of SDT.

Our starting point is the current structure of the current Advanced Computing track. In order to be admitted, applicants must hold an undergraduate degree in computer science or similarly strong technical background.

Programme Structure

The structure of the programme is depicted by the following table:

1st Semester	**Algorithm Design** (7.5 ECTS)	**Practical Concurrent and Parallel Programming** (7.5 ECTS)	**Elective Course or Project** (7.5 ECTS)	**Specialisation Course 1** (7.5 ECTS)
2nd Semester	**Advanced Programming** (7.5 ECTS)	**Security** (7.5 ECTS)	**Elective Course or Project** (7.5 ECTS)	**Specialisation Course 2** (7.5 ECTS)
3rd Semester	**Research Project** (7.5 ECTS)	**Elective Course or Project** (7.5 ECTS)	**Interdisciplinary Project** (7.5 ECTS)	**Specialisation Course 3** (7.5 ECTS)
4th Semester	**Master Thesis** (30 ECTS)			

We classify study activities into four different groups, each denoted by a different colour:

- Red activities are mandatory activities;

- Yellow activities are specialisation activities;
- Green activities are study activities the student can choose (electives or projects);
- The interdisciplinary project across ITU MSc programmes is in orange.
- The research project activity, a prerequisite for the Thesis, is purple.
- The MSc. thesis is a blue activity.

By default, all courses have 7.5 ECTS, however, green activities can be merged into bigger ones when possible, e.g., 15 ECTS courses or 15 ECTS projects.

Mandatory study activities. There are four main courses that each student must take. *Algorithm Design* is a course where students learn techniques for designing algorithms that efficiently solve certain problems. Practical Concurrent and Parallel Programming is a course that focuses on programming systems that make use of concurrency and parallel architectures. Advanced Programming is a course about advanced techniques for functional programming. Finally, Security is a course that teaches the principles of security for computer systems.
We identify these courses to be the basic knowledge for any graduate with an MSc. in Computer Science. All of them focus on advancing technical aspects of programming that we envision industry needs for.
Note: we are considering whether mandatory activities can be placed differently in the first two semesters, e.g., Security or Advanced Programming could be moved to the first semester.
Note 2: we are considering whether we should have a fifth mandatory course on processes for software development or theoretical computer science.

Specialisation Activities. Each student must choose a specialisation upon joining the programme (this can be changed later). A specialisation consists of three courses. The specialisation courses aim at giving a deep knowledge in one particular relevant area of computer science. Available specialisations are discussed in the next section.

Interdisciplinary Project Activity. This is a project that is intended to be done by groups made of students from different ITU MSc programmes. It is still not clear how this is going to be implemented. There is an idea that such project must apply some knowledge acquired in a course that runs in the previous semesters. Depending on the project, students can apply techniques learned in one of the four mandatory courses (all relevant). Computer Science students should definitely contribute with their technical skills.

Research Project. This project replaces the current Thesis Preparation. It will serve as a preliminary activity for getting students ready to start their thesis. It is strongly recommended that the project is connected to the specialisation taken by the student. It is crucial that passing this course is a prerequisite for being able to start a thesis. During the first 3-4 weeks, students will have to find a supervisor for the project. After that, they will work on the project. Examination consists of a project report to be handed in and an oral exam based on the report.

Specialisations

We currently suggest three main specialisation areas (names are imprecise and should be adjusted): Algorithms and Big Data, Software Engineering, and Systems. Each specialisation has a set of courses offered from which the student must select two courses (plus project).
Algorithms and Data. This specialisation should cover a classical specialisation on algorithms and/or business analytics. Current courses that could go in this specialisation: *Applied Algorithms, Intelligent Systems Programming, Big Data Management, Data Mining*. Suggested new courses: Machine Learning*?.*
Software Development. This specialisation should offer courses about software engineering from programming languages to software engineering. Current courses that could go in the specialisation: *Critical Systems and Security, Domain Specific Languages, Automated Software Analysis, Advanced Software Engineering, Software Architecture, Enterprise Architecture.* Suggested new courses: *none at the moment. s.*
Systems. This specialisation focuses on aspects of software that are closer to hardware. Current courses that could go in the specialisation: *Pervasive Computing, How to build (almost) anything, Mobile App Development, Big Data Management, Database Tuning.* Suggested new courses: *none at the moment.*

Target Group

In this section, we address the set of students that we wish to admit. In particular, we will analyse in detail what are the admission requirements and how applications should be structured.

Admission Requirements.

The requirements for being admitted to this programme are listed below:

- The applicant must hold a bachelor (or equivalent) in Computer Science or alike.
- The successful applicant is expected to have covered most of the following topics during his/her bachelor studies:
    - Propositional and predicate logic
    - Basic counting and probability theory
    - Sets and functions
    - Automata, grammars, and regular expressions
    - Algorithmic problem solving techniques
    - Sorting and shortest path algorithms
    - Trees and graphs
    - Hashing
    - big-O notation and complexity analysis
    - Programming language parsing techniques
    - Types
    - Object-Oriented Programming (including inner classes and generics)
    - A first exposure to threads and locks
    - Basic functional programming

Failure to cover such topics will penalise the applicant.

Application Structure.

Besides the formal requirements from the IT University, the applicant must submit a cover letter (max 2 pages) containing information describing how the list of requirements is met by the applicant (list of courses where you learned about the topics above).

Job Profiles

We expect that graduates from the Computer Science programme will have a job profile standard to all computer science graduates from most top universities, in Denmark and abroad. ITU is outsourcing a market research and we will analyse the data when available.