CURRICULUM FOR THE BACHELOR'S PROGRAMME IN SOFTWARE, 2020, COPENHAGEN

BACHELOR OF SCIENCE (BSC) IN ENGINEERING COPENHAGEN

MODULES INCLUDED IN THE CURRICULUM
# TABLE OF CONTENTS

- Analysis & Problem Formulation 2020/2021 .......................................................... 3
- A Program that Solves a Problem 2020/2021 ............................................................ 5
- Problem Based Learning 2020/2021 ........................................................................ 7
- Imperative Programming 2020/2021 ........................................................................ 9
- Datalogiens teoretiske grundlag 2020/2021 .............................................................. 11
- A Larger Program Developed by a Group 2020/2021 ............................................ 14
- Algorithms and Data Structures 2020/2021 ............................................................ 17
- Internetworking and Web-programming 2020/2021 ................................................. 19
- Probability Theory and Linear Algebra 2020/2021 .................................................. 21
- A Well-structured Application 2020/2021 ............................................................... 23
- Object-Oriented Programming 2020/2021 ............................................................... 26
- Systems Development 2020/2021 .......................................................................... 28
- Design and Evaluation of User Interfaces 2020/2021 ............................................. 30
- Design, Definition and Implementation of Programming Languages 2020/2021 .......... 32
- Languages and Compilers 2020/2021 ................................................................... 35
- Syntax and Semantics 2020/2021 .......................................................................... 37
- Computer Architecture and Operating Systems 2020/2021 ................................. 39
- Agile Software Engineering 2020/2021 .................................................................. 42
- Machine Intelligence 2020/2021 .......................................................................... 44
- Database Systems 2020/2021 ............................................................................ 46
- Bachelor's Project 2020/2021 ................................................................................ 48
- Algorithms and Computability 2020/2021 .............................................................. 50
- Security 2020/2021 ............................................................................................... 54
- Complex Front-end Software 2020/2021 ............................................................... 56
- Complex Back-end Software 2020/2021 ............................................................... 59
ANALYSIS & PROBLEM FORMULATION

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.

This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
That the student obtains knowledge of problem-oriented project work and specifically be able to analyze and define a problem within programs as well as describe the problem in an application context with emphasis on either a technical or social context.

REASONS
The project work focuses on gaining early experience of problem-oriented project work in groups to build competence for the P1 project.

CONTENTS
As documentation for the project work, the project team must:

• prepare a P0 report, and
• prepare a P0 process analysis

After submission of the project report, an experience session is held in which a number of P0 project groups present their experiences with the project group’s work process.

The experience session forms the basis for each group's process analysis.

LEARNING OBJECTIVES

KNOWLEDGE

• be able to understand and explain the theories and methods used in the project

SKILLS

• convey analysis and delimitation of a software problem
• communicate the project’s considerations, work results and work processes in writing, graphically and orally
• describe gained experiences with the group’s project work

COMPETENCES

• define a problem within software
• describe and analyze a problem and select and defend a problem formulation
• propose and argue for possible solutions to a formulated problem

TYPE OF INSTRUCTION

Project work

EXTENT AND EXPECTED WORKLOAD

Det forventes at den studerende bruger 27,5 timer per ECTS, hvilket for denne aktivitet betyder 137,5 timer.
EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Analysis &amp; Problem Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Oral exam based on a project</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>Passed/Not Passed</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Analyse &amp; problemformulering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB110</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
A PROGRAM THAT SOLVES A PROBLEM

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
That the student obtains skills in problem-oriented project work in a group as well as knowledge about the relationships between problem definition, the role of model development in understanding and constructing programs, and programs as a solution to a problem in a problem context. Furthermore, to gain knowledge of the subject content and the further potentials of the subject

LEARNING OBJECTIVES

KNOWLEDGE

• theories and methods for analyzing the chosen problem
• concepts in programming used in the project
• the contextual relationship of the project
• project management

SKILLS

• defining a problem leading to a problem formulation
• apply concepts and tools to problem-based project work
• select, describe and apply a method for organizing group collaboration and for resolving any group conflicts
• program a solution to a problem in collaboration with the other students in the group
• use collaboration tools to design and organize the group's documents and programs
• reflect on the problem-based learning in project context
• communicate the project's work results and work processes in a structured and understandable way, both in writing and orally

COMPETENCES

• analyze a problem within software
• formulate a problem where programming can be part of the solution
• include relevant concepts and methods for assessing project solutions in relation to the context of the problem
• participate in the development of a small, high-quality program
• describe and discuss the essential features of the program that contribute to the quality of the program

TYPE OF INSTRUCTION

Project work is supported by the courses on the semester and by digital learning resources in group and writing tools. In the middle of the project period, a status seminar will be held, where the project group will present problem formulation, work results and experience with the work process. After submission of the project report, an experience session is held in which a number of project groups present their experiences with the project work process. The experience session forms the basis for each group's process analysis

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which means 275 hours for this activity.
EXAM
EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>A Program that Solves a Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Oral exam based on a project</td>
</tr>
<tr>
<td>ECTS</td>
<td>10</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Et program, der løser et problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWC8121</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>10</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
PROBLEM BASED LEARNING

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

LEARNING OBJECTIVES

KNOWLEDGE

• central approaches, concepts and techniques in problem based learning
• different problem types, project types and their mutual relations
• theory of science in a problem based learning context

SKILLS

• define problem based learning with theory as well as own experiences as starting point
• plan and manage problem based project work taking into consideration the given problem type, team constellation and the duration of the project
• identify, analyse and formulate an open and complex problem considering its human and societal context
• point out relevant focus areas, concepts and methods to open and sustainable problem solving of complex problems
• discuss methodological impacts of different theory of science positions when working problem based
• analyse, compare and evaluate the processes of working with different problem types
• analyse and evaluate the processes of working team based in a problem based project, including project planning, monitoring and team development

COMPETENCES

• develop study practice aligned with a problem based, project organised and digitalised learning environment
• point out, try out and evaluate relevant techniques and approaches to improve a problem based project work
• transfer own experiences from a problem based project to guidelines for comparable projects
• evaluate own progression in problem based learning, based on experience and learning theory

TYPE OF INSTRUCTION

See § 17: Structure and content of the programme

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Problem Based Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>Passed/Not Passed</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>
FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Problembaseret læring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>TECHENGPBL20</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Aalborg, Campus Copenhagen, Campus Esbjerg</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Jette Egelund Holgaard</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Planning and Surveying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Planning</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
IMPERATIVE PROGRAMMING

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
In this course, the student will gain insight into basic concepts such as algorithms, data structures and computer architectures

REASONS
Computers, regardless of subject area, are one of the most important problem-solving tools today. Therefore, the student must acquire a knowledge of basic computer science concepts so they will be able to solve problems using imperative programming languages.

LEARNING OBJECTIVES

KNOWLEDGE

• Development environment and compilation
• Imperative principles
• Data types and variables
• Control structures
• Functions and procedures
• Data structures including arrays
• Input / Output
• Composite data structures
• Simple algorithms (eg sorting and searching)
• Basic testing of programs

SKILLS

• write, run and test programs incorporating the above-mentioned basic concepts into the solution
• apply correct subject terminology

COMPETENCES

both independently and in collaboration with others, implement an imperative program as a solution to a defined task

TYPE OF INSTRUCTION

Teaching is organized in accordance with the general teaching methods of the education, cf. section 17

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours
## EXAM

### EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Imperative Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

## ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

## FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Imperativ programmering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB105</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

## ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
DATALOGIENS TEORETISKE GRUNDLAG

2020/2021

MODULETS INDHOLD, FORLØB OG PÆDAGOGIK

FORMÅL
At den studerende tilegner viden, færdigheder og kompetencer inden for grundlæggende matematiske discipliner, som er væsentlige i mange datalogiske sammenhæng. Formålet er endvidere at bibringde disse færdigheder på en måde, som har en klar relevans, og tydelige anvendelser, inden for algoritmer, datastrukturer og programudvikling.

BEGRUNDELSE
Der er mange fagligheder i datalogi/software som kræver et solidt matematisk fundament.

LÆRINGSMÅL

VIDEN

- Mængdelære
  - Begreberne mængde, element i mængde; notation, Venn diagram
  - Mængdebyggernotation
  - Inklusion, lighed mellem mængder, tom mængde, potensmængde, kartesisk produkt
  - Mængdeoperationer
    - Foreningsmængde, fællesmængde, komplementærmængde
  - Mængderne af naturlige tal, rationelle tal og reelle tal
- Relationer
  - Binære relationer, transitiv afslutning
  - Ækvivalensrelationer (refleksiv, symmetrisk, transitiv), ækvivalensklasser, klassedeling
  - Ordningsrelation (antisymmetri)
- Funktioner
  - Definitionsmængde, værdimængde
  - Injektiv, surjektiv, bijektiv funktion
  - Specielle funktioner
    - Polynomier, exponential- og logaritme-funktioner
- Udsagnslogik
  - Basal definition af operatorer
  - Ækvivalens mellem udsagn, distributive love, de Morgans love, m.m.
  - Sandhedstabeller
  - Konjunktiv og disjunktiv normalformer
- Prædikatlogik
  - Prædikater, alkvantor, eksistenskvantor
  - Ækvivalens, de Morgans love
  - Normalformer i prædikatlogik
- Bevisteknikker
  - Bevis ved modstrid, slutningsregler
  - Bevis ved induktion på de naturlige tal
- Grundlæggende talteori
  - Divisibilitet, modulær aritmetik, Euklids algoritme
- Grundlæggende kombinatorik
  - Permutationer, kombinationer, binomialformlen
- Store-O notation med anvendelser på tidskompleksitet
- Rekursion og induktion
  - Rekursiv definition af funktioner
  - Rekursiv definition af mængder og simple strukturer: strenge og træer
  - Ræsonnementer om rekursivt definerede strukturer, strukturel induktion
FÆRDIGHEDER

• kunne gennemføre beviser for resultater indenfor kursets emner ved hjælp af de i kurset behandlede bevisteknikker
• kunne estimere tidskompleksiteten af simple algoritmer
• kunne gøre brug af de fornødne skriftlige færdigheder i disse sammenhænge

KOMPETENCER

• kunne identificere og anvende relevante matematiske teorier og teknikker, som er nødvendig til analyse af virkelige problemer
• kunne anvende matematiske teknikker som basis for udvikling af algoritmiske løsninger af et givet problem

UNDERVERSINGSFORM

Undervisningen tilrettelægges i henhold til de generelle undervisningsformer for uddannelsen, jf. § 17

OMFANG OG FORVENTET ARBEJDSINDSATS

Kursusmodulets omfang er 5 ECTS svarende til 137,5 timers studieindsats.

EKSAMEN

PRØVER

<table>
<thead>
<tr>
<th>Prøvens navn</th>
<th>Datalogiens teoretiske grundlag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prøveform</td>
<td>Skriftlig eller mundtlig</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Bedømmelsesform</td>
<td>7-trins-skala</td>
</tr>
<tr>
<td>Censur</td>
<td>Intern prøve</td>
</tr>
<tr>
<td>Vurderingskriterier</td>
<td>Vurderingskriterierne er angivet i Universitetets eksamensordning</td>
</tr>
</tbody>
</table>

YDERLIGERE INFORMATIONER

FAKTA OM MODULET

<table>
<thead>
<tr>
<th>Engelsk titel</th>
<th>The Theoretical Foundations of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulkode</td>
<td>B-MAT1-DTGC</td>
</tr>
<tr>
<td>Modultype</td>
<td>Kursus</td>
</tr>
<tr>
<td>Varighed</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Efterår</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Undervisningssprog</td>
<td>Dansk</td>
</tr>
<tr>
<td>Tomplads</td>
<td>Ja</td>
</tr>
<tr>
<td>Undervisningssted</td>
<td>Campus København</td>
</tr>
<tr>
<td>Modulansvarlig</td>
<td>Morten Grud Rasmussen</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
</tr>
</tbody>
</table>

**ORGANISATION**

<table>
<thead>
<tr>
<th>Studienævn</th>
<th>Studienævn for Matematiske Fag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Institut</td>
<td>Institut for Matematiske Fag</td>
</tr>
<tr>
<td>Fakultet</td>
<td>Det Ingeniør- og Naturvidenskabelige Fakultet</td>
</tr>
</tbody>
</table>
A LARGER PROGRAM DEVELOPED BY A GROUP

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer
This is an English translation of the module.

In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
That the student learns how to use the programming and associated model formation in problem solving. The aim is also to gain additional experience in problem-oriented project work in a group in natural progression from 1. Semester

JUSTIFICATION
The project work focuses on acquiring skills in programming and related modelling for problem solving. Project work must thus ensure that students achieve a common foundation of programming that can be used in following semesters. Project work must incorporate essential concepts about algorithms, data structures and/or computer networks. The project work also focuses on gaining experience with problem-oriented project work related to programming

LEARNING OBJECTIVES

KNOWLEDGE

• Algorithmic concepts relevant to project problem solving
• Distributed solutions in computer networks relevant to project problem solving
• concepts, structures and facilities in the programming language used
• Testing of applications, including unit testing

SKILLS

• Identify and define a problem within a given area

• Designing a well-structured solution

• Implement a larger, high-quality programme

• Incorporate concepts and techniques in the field of algorithmic and computer networks that are relevant to the project;

• Test the program to ensure that the program is correct

• Use version control to manage the application's parts in the development process
  • be able to understand and explain the concept of project management
COMPETENCES

• Identify a problem and, within this issue, formulate and defend a problem that can be solved by the use of programming;

• Prepare a model that can be used to solve the selected problem

• Complete the work to a running and demonstrable programme

• Describe how the program solves a formulated problem within the context of the problem

• Integrating relevant social and social aspects of solutions in the field of computer science and software technology;

• Describe, reflect on and analyse the experience gained with problem-oriented project work in a group

TYPE OF INSTRUCTION

Project work supported by the courses on the semester and digital learning resources in test, version control and program design. In the middle of the project period, a status seminar is held where the project team presents its problem formulation, work results and experience with the workflow

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 412.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>A Larger Program Developed by a Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Oral exam based on a project</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for computer science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Et større program udviklet af en gruppe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB220</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

**ORGANISATION**

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
ALGORITHMS AND DATA STRUCTURES

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

The student must gain knowledge of the following theories and methods:

• Mathematical basic concepts such as recursion, induction, concrete and abstract complexity

• Internal and external data structures, algorithm principles such as search, search trees, internal and external sorting, dynamic programming, part-and-intake

• Graphs and graph algorithms such as shortest road, consistency components, unfolding tree

SKILLS

• Determine abstract complexity for concrete features implement complexity and correctness analysis on simple algorithms, including recursive algorithms

• Select and use appropriate algorithms for standard tasks, such as search, sorting and finding

COMPETENCES

The student must, faced with a non-standard programming task, be able to

• Develop algorithms and data structures to solve the task

• Analyze the developed algorithms

TYPE OF INSTRUCTION

The training shall be organised according to the general teaching forms referred to in § 17

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.
## EXAM  
### EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Algorithms and Data Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

## ADDITIONAL INFORMATION

Kontakt: Studienævn for datalogi via cs-sn@cs.aau.dk eller 9940 8854

## FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Algoritmer og datastrukturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB221</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

## ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
INTERNETWORKING AND WEB-PROGRAMMING

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

The student must gain knowledge of the following theories and methods:

• Internetwork
  ◦ Principles for building computer networks; their stratification and structuring
  ◦ Knowledge of common network technologies
  ◦ Internet protocols, especially IP/UDP/TCP/HTTP
  ◦ Internet infrastructure services
  ◦ Security protocols
  ◦ Network programming, especially using sockets

• WEB programming
  ◦ Client-server architectures
  ◦ Web technologies, markup and scripting languages
  ◦ Development and use of web apis (e.g. rest)
  ◦ Testing techniques for web applications

• One or more topics of the following:
  ◦ Managing network failures
  ◦ Basic concurrency and communication with messages
  ◦ Design patterns for web applications
  ◦ Principles for distributed calculation (master-worker, pipelines, parallel algorithms,...)
  ◦ Protocol design principles
  ◦ Security in web applications

SKILLS

• Be able to accurately describe and use the terminology and notation for Internet work and web programming of the profession;
• Be able to explain the building of Internet networks and the functioning of key protocols
• Be able to program basic web applications

COMPETENCES

• Be able to apply concepts and techniques from Internet work and web programming

• Be able to develop basic internet-based applications

TYPE OF INSTRUCTION

The teaching is organised according to the general teaching forms referred to in § 17.
EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Internetworking and Web-programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for computer science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Internetværk og web-programmering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB212</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
PROBABILITY THEORY AND LINEAR ALGEBRA

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

In order to motivate the students, realistic examples from computer science are used to support the topics. The course includes the use of a programming environment so that the methods can be used.

LEARNING OBJECTIVES

KNOWLEDGE

• Grundlæggende sandsynlighedsteori
  ◦ Kombinatorik
  ◦ Sandsynlighedsteori for diskrete udfaldsrum
  ◦ Typer af fordelinger (Bernoulli, Binomial, Uniform,…)

• Grundlæggende lineær algebra
  ◦ Vektorrum i Rn
  ◦ lineære transformationer og deres sammenhæng med matricer
  ◦ simple matrixoperationer
  ◦ invertibel matrix og invertibel lineær afbildning
  ◦ ortogonale projektioner

• Mindste kvadraters lineær regression

• Lineær programmering

• Diskretids Markov kæder
  ◦ sandsynligheden for opnåelige tilstande
  ◦ stationære fordelinger

SKILLS

• be able to apply the least squares method to linear regression problems
• represent systems of linear equations by matrix equations and use Gaussian elimination to solve them
• determine matrix for a given linear plot, and vice versa
• solve simple matrix equations
• calculate inverse of small matrices
• calculate the orthogonal projection of a vector on a subspace of Rn (Gram-Schmidt algorithm)

COMPETENCES

• develop and strengthen knowledge of, understanding of, and application of mathematical theories and methods within other subject areas based on given assumptions
• be able to reason and argue with mathematical concepts within probability theory and linear algebra

TYPE OF INSTRUCTION

The teaching is organized in accordance with the general teaching methods for the education, cf. section 17.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.
EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Probability Theory and Linear Algebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Sandsynlighedsteori og lineær algebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB222</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
A WELL-STRUCTURED APPLICATION

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
The module builds on knowledge, skills and competences gained in the project modules in the 1st - 2nd semester

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

• analyze and model requirements in the object-oriented paradigm

• design a software architecture using common design patterns

• design, program and test an application in the object-oriented paradigm

• understand and utilize concepts and facilities in both object-oriented analysis, design and programming and, on this basis, construct an application of high, internal and external quality

SKILLS

• perform systematic testing of the application as a whole and demonstrate that the application matches the intentions and users’ needs

• carry out systematic testing of an object-oriented program with special emphasis on integration testing

• carry out systematic user interface evaluation

• argue for choices made in all activities of the development process, including explaining requirements, architecture and how users’ needs are interconnected

• gain skills in balancing multiple learning goals

COMPETENCES

• develop a running application that solves user problem
• apply object-oriented programming

• describe and reflect on the working method used in the development project

TYPE OF INSTRUCTION

Project work including:

• formulation and analysis a problem whose solution can be described in the form of analysis, design and realization of a specific application
• The problem should be found outside the university and be analyzed based on the involvement of users, clients, or customers
• elaborations of such a solution
• reflection on this process of development

The project work is supported by digital resources on integration testing and writing competence in technical documentation and scientific dissemination.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 412.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>A Well-structured Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Oral exam based on a project</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>En velstruktureret applikation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB310</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>
## ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
OBJECT-ORIENTED PROGRAMMING

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
The module builds on the knowledge gained in the modules Programming, and Algorithms and Data Structures.

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
That the student learns the essential concepts and structuring mechanisms within object-oriented programming languages and acquires skills in programming in a language within this paradigm.

REASONS
Object-oriented programming is a dominant programming paradigm in software development.

LEARNING OBJECTIVES

KNOWLEDGE
The student must gain knowledge of concepts within the object-oriented programming paradigm and architecture, including a selection of the following:

- concepts and concept formation in object-oriented programming
- classes, methods, interfaces, abstract classes and generic variants
- scope rules and visibility
- instance and class members
- inheritance
- exceptions
- polymorphism
- encapsulation and abstraction
- standard library object-oriented design, architecture and principles
- design patterns
- specialization, implementation and delegation
- structured test
- documentation

SKILLS

- Construct programs that demonstrate the object-oriented principles of design and architecture covered in the course
- apply the concepts of classes, interfaces, objects, methods and polymorphism
- reason about program design and design patterns, and explain contexts in an object-oriented program
- prepare systematic testing of an object-oriented program
- apply correct terminology

COMPETENCES

- Design, document and test a large object-oriented program, and use the architectural concepts used in the course
- analyze and discuss a program based on object-oriented design principles and concepts
- define and discuss key concepts in object-oriented programming
**TYPE OF INSTRUCTION**

The teaching is organized in accordance with the general teaching methods for the education, cf. section 17

**EXTENT AND EXPECTED WORKLOAD**

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

**EXAM**

**EXAMS**

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Object-Oriented Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

**ADDITIONAL INFORMATION**

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

**FACTS ABOUT THE MODULE**

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Objektorienteret programmering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB311</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

**ORGANISATION**

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
SYSTEMS DEVELOPMENT

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

The student should gain knowledge of the following theories and methods:

Object-oriented modelling in analysis and design:
• modelling of context (application domain and problem domain)
• object-oriented concepts: class, object, event, structure, function, use patterns, component, component architecture
• UML: class diagram, state chart diagram, sequence diagram, diagram for use patterns

Modelling with patterns:
• patterns for modelling application and problem domains
• patterns for composing components
• specifically the patterns for analysis: object-descriptor, hierarchy, stepwise-role, materials, procedure
• specifically the patterns for design: collection, layered, observer, client-server, model-view-controller

System development methods:
• waterfall method and model-driven development
• iterative method and prototype-driven development
• activities in systems development and relations between activities

Systems practices:
• the relation between methodology and practice
• strengths and weaknesses of model-driven and prototype-driven development

SKILLS

• be able to explain accurately, using the concepts and modelling language of the discipline

• be able to model the requirements to a system, its context and all its various parts (model, features and interfaces)

• be able to model a system design at component level and describe relations between components

COMPETENCES

• be able to apply concepts, patterns and modelling language to describe a specific system that solves a well-defined task

TYPE OF INSTRUCTION

The type of instruction is organised in accordance with the general instruction methods of the programme, cf. § 17.
EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Systems Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: The Study board for Computer Science at cs-en@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Systemudvikling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB312</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
DESIGN AND EVALUATION OF USER INTERFACES

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE

The module builds on knowledge gained in the project modules in the 1st and 2nd semester.

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

The student must gain knowledge of the following theories and methods:

Fundamentals of human-computer interaction:
- interaction design
- usability and user experience
- design principles
- interaction forms

Interaction design process:
- activities in interaction design
- user-centered design

Use context and users:
- understand needs and requirements: e.g. interview, observation, questionnaire, probes, card sorting
- task analysis: e.g. hierarchical task analysis, objectives, tasks, actions
- scenarios and personas
- use patterns

Design of interfaces:
- visual design principles
- Gestalt laws
- sketching and prototyping
- conceptual and physical interface design

Usability evaluation:
- activities
- roles and tasks
- identification of usability problems

SKILLS

- understand basic and advanced concepts and theories of human-computer interaction
- be able to explain the activities in the design of an interface accurately
- be able to explain the activities of a usability evaluation
COMPETENCES

- be able to apply concepts, techniques and methods to design and evaluate a specific system that solves a well-defined task

TYPE OF INSTRUCTION

The teaching is organized according to the general teaching methods for the education, cf. §17

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Design and Evaluation of User Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Design og evaluering af brugergrænseflader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB313</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
DESIGN, DEFINITION AND IMPLEMENTATION OF PROGRAMMING LANGUAGES

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
The module builds on knowledge from the project modules on 1-3. term. It is recommended that the modules Syntax and Semantics, Language and Translators and Computer Architecture and Operating Systems be followed in parallel with the project work.

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
That the student learns how to design and implement a programming language and how this process can be supported by formal definitions of the language's syntax and semantics and techniques and methods for translator and / or interpreter construction.

REASON
All software is written in a programming language and is translated or interpreted in order to be executed. Design, description and construction of Programming languages, translators, interpreters and similar tools are therefore central topics in computer science.

Therefore, the project work focuses on ensuring that the students understand important underlying concepts in the world of programming languages, why these concepts have arisen and how they are formally described and represented in an implementation. Understanding these topics is fundamental in understanding new and existing programming languages and their applications.

Furthermore, many techniques and tools, originally developed for languages and translators, are also used in other contexts in program development.

LEARNING OBJECTIVES

KNOWLEDGE

• understand and explain the basic concepts in a definition of the programming syntax and semantics of a programming language
• document knowledge of and overview of the relevant techniques and concepts in language design and translator construction
• explain the individual phases and the connection between the phases in a translator or interpreter
• explain the implementation techniques used in the constructed translator / interpreter
• use correct subject terminology

SKILLS

• describe the syntax and semantics of a programming language using relevant methods
• implement a translator or interpreter for a specific programming language or for an extension to an existing programming language
• test the implemented translator or interpreter at all levels: unit, integration and acceptance test
• explain configuration management during the development of translator or interpreter
• reason computerologically about and with the concepts and techniques concerned

COMPETENCES

• assess the use and applicability of known tools and techniques for the definition and implementation of programming languages
• understand and explain how concrete linguistic concepts are represented at driving times and in formal semantics

TYPE OF INSTRUCTION

Project work

The project is based on an analysis of a computer science problem, the solution of which can naturally be described in the form of a design of a specific programming language. The project includes a formal definition of relevant parts of this programming language. The focus of the project is the construction of a robust translator / interpreter for key parts of the language

Digital support for competence development within configuration management

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 412.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Design, Definition and Implementation of Programming Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Oral exam based on a project</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Kontakt: Studienævn for datalogi via cs-sn@cs.aau.dk eller 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Design, definition og implementation af programmeringssprog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB410</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
</tbody>
</table>
### Empty-place Scheme
- **Yes**

### Location of the lecture
- **Campus Copenhagen**

### Responsible for the module
- **Lone Leth Thomsen**

## ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
LANGUAGES AND COMPILERS

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE

The module builds on the knowledge gained in the modules Programming, Object Oriented Programming and the project module at 3. Semester.

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

The student must gain knowledge of essential principles in programming languages, as well as understanding of techniques for describing and translating languages in general, including:

• The abstraction principle, control and data structures, explicit and implicit execution order, block structure and scope concept, parameter mechanisms, types and type of equivalence;

• Translation, including lexical, syntactical, and static semantic analysis, and code recognition

• Run-time environment, including data representation, storage allocation, and sub-program support structures: Methods, procedures, and functions

SKILLS

• Be able to explain the techniques and concepts involved in language design and translation design using the terminology of the course and notation for the description and implementation of programming languages;

• Be able to explain how implementations techniques influence language design

• Be able to reason about and with the concepts and techniques involved

COMPETENCES

• Be able to describe, analyse and implement programming languages

• Be able to explain the individual phases and the correlation between the phases of a translator

TYPE OF INSTRUCTION

The teaching is organised according to the general teaching forms referred to in § 17.
EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Languages and Compilers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Practical exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for computer science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Sprog og oversætttere</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB411</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
SYNTAX AND SEMANTICS

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
The module builds on the course modules "The Theoretical Foundations of Computer Science" and "Algorithms and Data Structures"

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE
The student must gain knowledge of the following theories and methods:

Formal language theory:
- Theories of description and recognition of regular languages: deterministic and nondeterministic final machines, regular expressions and equivalence of these
- Theories of description and recognition of context-free languages: Context-free grammatics and pushdown machines and the equivalence of these
- Limitations of regular and context-free languages: Pumping Lemma for regular and context-free languages

Semantics of programming languages:
- Structural operational semantics: Big-step and small-step semantics of common programming constructions. Semantic equivalence. Semantics of scope rules and parameter mechanisms
- Techniques of program correctness. E.g. Hoare logic, type of systems
- Recursive definitions and calculation of fixed points

SKILLS
- Be able to accurately describe and use the terminology and notation of the course for results in formal language theory and semantics of programming languages, and
- how and to what extent these results can be used
- Be able to make use of the necessary written skills in these contexts.
COMPETENCES

- Be able to apply concepts and techniques from formal language theory and semantics of programming languages, including in the design and description of programming languages.

TYPE OF INSTRUCTION

The teaching is organised according to the general teaching forms cf. §17

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Syntax and Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Kontakt: Studienævn for datalogi via cs-sn@cs.aau.dk eller 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Syntaks og semantik</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWC8412</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
COMPUTER ARCHITECTURE AND OPERATING SYSTEMS
2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
The module builds on the knowledge gained in the modules "Programming" and "Internetworking and web-programming"

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

Give understanding of how applications run on a computer in conjunction with system software, utilize this to streamline applications, as well as introduce multi-programming.

Programs are written in high-level languages, translated into machine code and then driven on computers in conjunction with its operating system. The course gives the student a knowledge of the various steps in translating and executing high-level programs on computers, including how a specific processor executes a program.

The course also introduces how the architecture and system services can be used in application programmes with a focus on concurrency.

In this course, the student obtains knowledge of basic concepts in computer architecture, operating systems, and concurrency.

LEARNING OBJECTIVES

KNOWLEDGE

• Abstraction layers in computer architecture and operating systems
• Organization of hardware components (processor, memories, peripherals, buses,...)
• Processor architectures (Harvard, von Neuman) and types (embedded, desktop, high-performance, server, micro-controllers, DSP)
• Number and data representation.
• Instruction set architectures
• Instruction level parallelism
• Digital logical circuits and Boolean algebra
• The inventory hierarchy
• Virtual memory
• The driving time environment of a running program
• Interruptions (interrupts), system calls, and exceptions
• Cores and operating systems, virtual machines
• Multi-programming: processes and threads, synchronisation, deadlocks
• Basic multi-core programming

SKILLS

• Be able to explain the behavior of simple application at the assembler level

• Be able to understand and use special instructions in programs
• Be able to write programs to make the best use of your computer's hardware and processor, such as the storage hierarchy, instructional level parallelism, and explicit parallelism

• Be able to analyze and write simple, system-close programs that use parallelism and/or concurrency

• Be able to use the techniques of the profession to ensure reciprocal exclusion, fairness and absence of deadlock in simple simultaneous/parallel systems

• Be able to use correct terminology, accurately describe and use the terminology and notation of the course for building, structuring, functionality and operation of the computer architecture and operating systems

COMPETENCES

• After the course, the student can formulate and solve simple low-level-related issues that are grounded in knowledge of computer architecture, running time environments, operating systems, etc. - both independently and in cooperation with others.
• By synthesis of the profession's concepts and techniques, the student should be able to develop system-close simple programs that use parallelism and/or simultaneity.
• The student must be able to acquire new knowledge of operating systems and programming of simultaneous and parallel systems

TYPE OF INSTRUCTION

The teaching is organised according to the general teaching forms referred to in § 17.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Computer Architecture and Operating Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Computerarkitektur og operativsystemer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB413</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

**ORGANISATION**

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
AGILE SOFTWARE ENGINEERING

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

• leading paradigms in professional software development (agile development and plan-driven development)
• especially with a focus on agile software engineering, examples of different methods and techniques as well as the theories behind
• agile development in large projects ('agile in the large')
• process models, requirements management, design, project management, testing, process improvement

SKILLS

• be able to explain precisely and using the terminology of the subject the selected paradigms, and be able to distinguish and compare these
• be able to explain precisely and using the terminology of the subject for theories, methods and techniques within agile and plan-driven software engineering and their application in professional development of software intensive systems

COMPETENCES

• be able to select, justify and apply appropriate theories, methods and techniques in their own development projects

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Agile Software Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: The Study board for Computer Science at cs-sn@cs.aau.dk or 9940 8854
FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Agil software engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB512</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
MACHINE INTELLIGENCE

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
It is recommended that the student has knowledge of discrete mathematics, algorithms and data structures

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

The student should gain knowledge of the following theories and methods:

• problem solving using search and inference
• model-based decision making
• inference under uncertainty
• learning from experience and learning from data

SKILLS

• use correct technical notation and terminology in writing as well as speech

• apply basic techniques presented in the course to solve a specific problem

• explain key principles and algorithms presented in the course

COMPETENCES

• be able to evaluate, compare and select techniques and methods within machine intelligence based on a specific problem

TYPE OF INSTRUCTION

The type of instruction is organised in accordance with the general instruction methods of the programme, cf. § 17.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

| Name of exam | Machine Intelligence |
Type of exam | Written or oral exam
ECTS | 5
Assessment | 7-point grading scale
Type of grading | Internal examination
Criteria of assessment | The criteria of assessment are stated in the Examination Policies and Procedures

ADDITIONAL INFORMATION

Contact: The Study board for Computer Science at cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

| Danish title | Maskinintelligens |
| Module code | DSNSWCB513 |
| Module type | Course |
| Duration | 1 semester |
| Semester | Autumn |
| ECTS | 5 |
| Language of instruction | Danish and English |
| Empty-place Scheme | Yes |
| Location of the lecture | Campus Copenhagen |
| Responsible for the module | Lone Leth Thomsen |

ORGANISATION

| Study Board | Study Board of Computer Science |
| Department | Department of Computer Science |
| Faculty | Technical Faculty of IT and Design |
DATABASE SYSTEMS

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE

The module builds on knowledge gained in the module Algorithms and Data Structures

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer

This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

The student must gain insight into the following topics in database systems:

- the relational model and relational algebra entity relationship charts (ERD)
- Structured Query Language (SQL)
- logical design of relational databases (normal forms)
- physical database design
- query management and optimization
- transactions, concurrency control and recovery

A number of possible topics that can also be covered in the course depending on the teacher's and the students' abilities and background, for example:

- relational calculus
- parallel databases
- distributed databases
- triggers and stored procedures

SKILLS

- be able to explain the relational model and apply relational algebra to a data set
- be able to construct an ERD for smaller, concrete scenarios
- be able to create a relational database design that adheres to recognized normal forms
- be able to use SQL to create and query a database
- be able to explain an execution plan for an SQL query and assess whether the plan is effective
- be able to explain the concept of transaction and key topics within concurrency control and recovery

COMPETENCES

Using the fundamental concepts and theories that are common to most database systems, the student must be able to use these to handle larger data sets in practice.

TYPE OF INSTRUCTION

The teaching is organized in accordance with the general teaching methods for the education § 17.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.
EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Database Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Kontakt: Studienævn for datalogi via cs-sn@cs.aau.dk eller 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Databasesystemer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB514</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
BACHELOR'S PROJECT

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE

Builds on knowledge gained on the project and course modules on 1-5. semester

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer

This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE

University degrees are research-based degrees; At the end of the bachelor's program, all students must have gained insight into an aspect of the subject's research area. The theme for the semester is cyber physical systems, ie systems that contain both embedded software and a physical component that often involves an interaction directly with people. It is possible to treat Cyber Physical Systems from different research perspectives

LEARNING OBJECTIVES

KNOWLEDGE

• document knowledge of and overview of the relevant techniques and concepts within a computer science research area
• use correct subject terminology

SKILLS

• reason about and with the concepts and techniques concerned
• justify and choose relevant solution models based on knowledge of the possibilities and limitations given by the subject area's theories and methods
• convey a computer science problem and the associated conceptual apparatus

COMPETENCES

• apply the concepts and reasoning within the subject area to analyze and solve a selected problem within the chosen computer science subject area
• demonstrate a coherent understanding of the project's relationship to the subject area

TYPE OF INSTRUCTION

Project work supported by digital resources on knowledge search.

The project work must include:

• an analysis of a computer science problem and a formulation of a problem within it
• a solution and perspective of this problem

The project report can be designed as a scientific article of form and length

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 412.5 hours.
**EXAM**

**EXAMS**

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Bachelor's Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Master's thesis/final project</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

**ADDITIONAL INFORMATION**

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

**FACTS ABOUT THE MODULE**

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Bachelorprojekt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB610</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

**ORGANISATION**

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
ALGORITHMS AND COMPUTABILITY

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
The module builds on knowledge gained in modules: The Theoretical Foundations of Computer Science, Algorithms and Data Structures and Syntax and Semantics.

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE
The student must gain knowledge of the following theories and methods:

- algorithm design techniques such as share-and-rule, greedy algorithms, dynamic programming, backtracking, branch-and-bound algorithms, randomized algorithms and linear programming
- techniques in advanced algorithm analysis such as amortized analysis, analysis of expected complexity and experiments with algorithms
- examples of core algorithms and data structures for solving a number of problems from different areas of computer science such as algorithms for external memory, multi-threaded algorithms, search in text, advanced graph algorithms and geometric calculations
- decisive and recognizable decision-making problems; Turing machines; Church-Turing thesis; examples of reduction
- complexity theory; deterministic and nondeterministic Turing machines and their time complexity; time complexity classes; polynomial reductions; NP completeness; examples of NP-complete problems

SKILLS

- be able to account accurately and using the subject's terminology and notation for important results within calculation complexity and for how and to what extent these results can be used to classify calculation problems, including complexity analysis and applications of reductions
- explain the principles behind the most important algorithm design and analysis techniques
- select and apply algorithm design and analysis techniques for a given problem
- recognize a range of problems from different areas of computer science and select the most appropriate algorithms and data structures to solve them

COMPETENCES

- be able to apply concepts and techniques within computational complexity to analysis of computational problems
- when facing a computer science problem, be able to develop and analyze efficient algorithms and data structures for solving the problem

TYPE OF INSTRUCTION

The teaching is organized in accordance with the general teaching methods for the education, cf. section 17.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.
**EXAM**

### EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Algorithms and Computability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

**ADDITIONAL INFORMATION**

Kontakt: Studienævn for datalogi via cs-sn@cs.aau.dk eller 9940 8854

**FACTS ABOUT THE MODULE**

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Algoritmer og beregnelighed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB611</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

**ORGANISATION**

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
MODELS AND TOOLS FOR CYBER PHYSICAL SYSTEMS

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
The module builds on the project and course modules on 1-5. semester

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
Cyber Physical Systems, systems that both contain embedded real-time software and consist of a physical component, are becoming more and more widely used. This course supports the semester project by presenting relevant methods and theories

LEARNING OBJECTIVES

KNOWLEDGE

- formal models for Cyber Physical systems
- tools for simulation and analysis of functional, time and performance requirements
- different languages for implementing real-time systems
- real-time operating systems
- embedded hardware
- data collection and communication
- dynamic and / or hybrid systems

SKILLS

- explain the special types of requirements that apply to Cyber Physical Systems
- explain the definitions of the formal models of Cyber Physical Systems reviewed in the course

COMPETENCES

- reason about the choice of technologies for the development of a Cyber Physical System.
- demonstrate a coherent understanding of the various components that make up a Cyber Physical System

TYPE OF INSTRUCTION

The teaching is organized in accordance with the general teaching methods for the education, cf. section 17.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.
## EXAM

### EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Models and Tools for Cyber Physical Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

## ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

## FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Modeller og værktøjer for Cyber Physical Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWC612</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

## ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
SECURITY
2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

LEARNING OBJECTIVES

KNOWLEDGE

• CIA model (basic security features)
• access control models, eg Multi-Level Security, Biba, Role-Based and Attribute-Based Access Control
• processes for secure software development, eg building security in, OpenSAMM or Secure Development Lifecycle (SDL)
• software security, eg language-based security, secure information flow and techniques / tools for securing software
• basic network security, including knowledge of basic analysis and attack tools
• basic attack methods ("hacking"), eg "reverse engineering", including knowledge of various attack tools

SKILLS

• be able to perform high-level security analyzes, eg using the CIA model, of a simple IT system
• be able to use one or more relevant security tools for analysis, modeling or (simulated) attacks
• be able to document and prioritize identified security features / problems in a smaller IT system

COMPETENCES

• be able to use one or more of the above models / theories to identify and analyze relevant security features in a small IT system; and on the basis of a security analysis be able to propose and argue for the choice of countermeasures as well as select / design an appropriate security model for a smaller IT system
• be able to keep up to date with the latest developments, especially in attack methods and targets as well as associated countermeasures

TYPE OF INSTRUCTION

The teaching is organized in accordance with the general teaching methods for the education, cf. section 17.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 137.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>External examination</td>
</tr>
</tbody>
</table>
Criteria of assessment | The criteria of assessment are stated in the Examination Policies and Procedures

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Sikkerhed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB613</td>
</tr>
<tr>
<td>Module type</td>
<td>Course</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Spring</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
COMPLEX FRONT-END SOFTWARE

2020/2021

PREREQUISITE/RECOMMENDED PREREQUISITE FOR PARTICIPATION IN THE MODULE
Builds on knowledge gained from the project and course modules on 1st-5th. semester

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
The student will gain knowledge and skills in the analysis, design, implementation and evaluation of complex software systems that incorporate user interface in a larger development environment

REASONS
After completing training in software technology, students should be able to participate in problem solving by developing complex software systems. Typically, development of such complex systems takes place in a larger development environment, where subprojects must be coordinated for a common solution.

The student must therefore gain an understanding of issues related to the development of such systems. Working on analyzing, designing, programming and testing an application with essential functionality, including both database integration and user interface that must form part of a complex technical or organizational environment, focuses on requirements analysis, requirements management, management between subprojects, prototyping, programming of large software systems, testing, and advanced application of software technologies.

During the semester, two themes are chosen, each with their own learning goals. These two themes are Front-end and Back-end.

Practically, the semester is expected to run so that several groups with both front-end and back-end focus together develop a larger IT system

LEARNING OBJECTIVES

KNOWLEDGE

• be able to document knowledge and overview of key techniques in the work of agile software development that solves realistic problems
  ◦ requirements analysis
  ◦ requirements management
  ◦ prototyping
  ◦ usability
  ◦ testing and verification

• use correct subject terminology

SKILLS

• analyze, design, program, test and test applications that are part of a complex organizational environment
• reason about and with the concepts and techniques involved
• justify and select relevant reading models based on knowledge of the possibilities and limitations given by the field's theories and methods
COMPETENCES

- delineate and implement part of a major software development problem using relevant techniques
- analyze and evaluate the solution process and the solution obtained
- apply project management

TYPE OF INSTRUCTION

Project work in multi-projects. Each group submits their own report, but collaborates across several groups on the development of a common system

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 412.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Complex Front-end Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Oral exam based on a project</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Kompleks Front-end Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB520</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>
COMPLEX BACK-END SOFTWARE

2020/2021

CONTENT, PROGRESS AND PEDAGOGY OF THE MODULE

Disclaimer.
This is an English translation of the module. In case of discrepancy between the translation and the Danish version, the Danish version of the module is valid.

PURPOSE
That the student obtains knowledge and skills in analysis, design, implementation and assessment of complex software systems in a larger development environment

REASONS
After completing training in software technology, students should be able to participate in problem solving by developing complex software systems. Typically, development of such complex systems takes place in a larger development environment, where subprojects must be coordinated for a common solution.

The student must therefore gain an understanding of issues related to the development of such systems. Working on analyzing, designing, programming and testing an application with essential functionality, including both database integration and user interface that must form part of a complex technical or organizational environment, focuses on requirements analysis, requirements management, management between subprojects, prototyping, programming of large software systems, testing, and advanced application of software technologies.

During the semester, two themes are chosen, each with their own learning goals. These two themes are Front-end and Back-end.

Practically, the semester is expected to run so that several groups with both front-end and back-end focus together develop a larger IT system.

LEARNING OBJECTIVES

KNOWLEDGE

• be able to document knowledge and overview of key techniques in the work of agile software development that solves realistic problems
  o requirements management
  o prototyping
  o databases
  o testing and verification
• use correct subject terminology

SKILLS

• analyze, design, program, test and test applications that are part of a complex technical context
• reason about and with the concepts and techniques concerned
• justify and select relevant reading models based on knowledge of the possibilities and limitations given by the field’s theories and methods

COMPETENCES

• delineate and implement part of a major software development problem using relevant techniques
• analyze and evaluate the solution process and the solution obtained
• apply project management
• collaboration in multi-project across groups
TYPE OF INSTRUCTION

Project work in multi-projects. Each group submits their own report, but collaborates across several groups on the development of a common system.

EXTENT AND EXPECTED WORKLOAD

The student is expected to spend 27.5 hours per ECTS, which for this activity means 412.5 hours.

EXAM

EXAMS

<table>
<thead>
<tr>
<th>Name of exam</th>
<th>Complex Back-end Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of exam</td>
<td>Oral exam based on a project</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Assessment</td>
<td>7-point grading scale</td>
</tr>
<tr>
<td>Type of grading</td>
<td>Internal examination</td>
</tr>
<tr>
<td>Criteria of assessment</td>
<td>The criteria of assessment are stated in the Examination Policies and Procedures</td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Contact: Study Board for Computer Science via cs-sn@cs.aau.dk or 9940 8854

FACTS ABOUT THE MODULE

<table>
<thead>
<tr>
<th>Danish title</th>
<th>Kompleks Back-end Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>DSNSWCB521</td>
</tr>
<tr>
<td>Module type</td>
<td>Project</td>
</tr>
<tr>
<td>Duration</td>
<td>1 semester</td>
</tr>
<tr>
<td>Semester</td>
<td>Autumn</td>
</tr>
<tr>
<td>ECTS</td>
<td>15</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>Danish and English</td>
</tr>
<tr>
<td>Empty-place Scheme</td>
<td>Yes</td>
</tr>
<tr>
<td>Location of the lecture</td>
<td>Campus Copenhagen</td>
</tr>
<tr>
<td>Responsible for the module</td>
<td>Lone Leth Thomsen</td>
</tr>
</tbody>
</table>

ORGANISATION

<table>
<thead>
<tr>
<th>Study Board</th>
<th>Study Board of Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>Department of Computer Science</td>
</tr>
<tr>
<td>Faculty</td>
<td>Technical Faculty of IT and Design</td>
</tr>
</tbody>
</table>