

# Computer Science Electives Spring 2026

In the spring of 2026, the following electives will be offered in Computer Science:

- DevOps
- Internet of Things (IoT)
- Fullstack Development
- Mobile Development
- Applications of Artificial Intelligence

## Examination

Exams are expected to take place in June.

<h2>DevOps</h2>
<p>Teacher: Thomas Boulund Meng Hansen          Language: English          This elective takes place by attendance</p>
<p><u>Contents:</u>          This course discusses the concept of DevOps as an approach to delivering software in a fast and secure manner without compromising quality and stability of the deliveries. Our primary tools to succeed in that journey will be automation through continuous integration, delivery and deployment combined with strategies to recover if something doesn't work out as expected.</p> <p>10 ECTS</p>
<p><u>Learning objectives</u></p> <p><b>Knowledge</b>          The student must have knowledge of:</p> <ul style="list-style-type: none"> <li>• The overall objective of DevOps and how working with automation, measurement and recovery can help achieve these problems</li> <li>• The concept of pipelines and how it relates to DevOps</li> <li>• Different aspects within containerisation and how these aspects relate to DevOps</li> </ul> <p><b>Skills</b>          The student can:</p> <ul style="list-style-type: none"> <li>• Implement a full DevOps pipeline in GitHub Actions from integration through delivery to deployment</li> <li>• Integrate different types of software testing automatically into the software development life cycle</li> <li>• Measure the performance of any software project</li> <li>• Monitor their applications and use technology to revert failing changes in production</li> </ul> <p><b>Competencies</b>          The student can:</p> <ul style="list-style-type: none"> <li>• Plan and automate processes to quickly deliver new features without compromising quality</li> <li>• Monitor their software solutions and implement a shift-left approach to detect and resolve bugs early in the development process</li> </ul>
<p><u>Prerequisites for the exam, exam form and organization, evaluation criteria and any report structure and requirement</u></p> <p><b>Formal requirements for the project</b>          Every week throughout the semester you're going to develop a project with additional requirements each week. After the last week of teaching you're going to hand-in a link to a public and archived GitHub repository.</p>

**Formal requirements for the written submission**

No written submission expected.

**The examination**

The mode of exam is "Known questions". After the last week of teaching a list of questions is handed out. During the exam period you're going to prepare presentations for each individual question.

At the day of the exam you are going to draw a random question. Without preparation time you're going to present that question.

Time perspective for the oral exam:

- **10 minutes:** Present the question - I might ask follow-up questions along the way.
- **5 minutes:** Voting and grading.

The exam is conducted online over Zoom.

**Assessment**

7-point grading scale. Grading is based on the product and oral examination.

<h2>Internet of Things (IoT)</h2>
<p>Teacher: Jeppe Led Language: English This elective takes place by attendance</p>
<p><u>Contents:</u> IoT aims at equipping students with the necessary concrete and abstract skills to be competent to develop prototypes of IoT systems, ultimately to explore new ways of creating value for business. 10 ECTS</p>
<p><u>Learning objectives</u></p> <p><b>Knowledge</b></p> <ul style="list-style-type: none"> <li>• Basic overview of the IoT eco-system and current usages</li> <li>• Understanding IoT's role in business</li> </ul> <p><b>Skills</b></p> <ul style="list-style-type: none"> <li>• Programming microcontrollers, including relevant programming languages and deployment</li> <li>• Connecting to different sensors e.g. temperature, air-pressure, humidity, rain, distance, etc.</li> <li>• Connecting to modules e.g. motors, fans, pumps, lights, etc.</li> <li>• Radio communication, e.g. Bluetooth, BLE, 433 Mhz</li> <li>• Digital vs. analog electronics in practical use and digital/analogue conversion</li> <li>• Basic understanding of driver software for different hardware</li> <li>• Decentralized communication protocols for small devices e.g. MQTT</li> <li>• Data collection from IoT devices</li> </ul> <p><b>Competences</b></p> <ul style="list-style-type: none"> <li>• Designing prototypes of IoT devices</li> <li>• Advising on core technical implementations of IoT in a business setting</li> <li>• Integrating custom IoT prototypes with other systems</li> </ul>
<p><u>Prerequisites for the exam, exam form and organization, evaluation criteria and any report structure and requirement</u></p> <p><b>The examination</b></p> <p>Exam is held as an individual, project and synopsis based, oral exam. The project is a group project. Both the synopsis and project scope, and required content, is defined under the course on the LMS.</p> <p>The exam is 20 minutes in total including grading and overhead.</p> <p>The student presents the project and show relevant parts that fulfill one or more of the knowledge, skills and competencies listed above.</p> <p><b>Assessment</b></p> <p>7-point grading scale.</p>

<h2>Fullstack Development</h2>
<p>Teacher:</p> <p>Language: English</p> <p>This elective takes place by attendance</p>
<p><u>Contents:</u></p> <p>The purpose of the course is to introduce the student to real-time communication systems between browsers and web servers and advanced software architecture, such that students can develop novel business applications using modern technologies.</p> <p>10 ECTS</p>
<p><u>Learning objectives</u></p> <p><b>Knowledge</b></p> <p>The student must have knowledge of:</p> <ul style="list-style-type: none"> <li>• Protocols and technologies used to facilitate real-time web communication</li> <li>• Code organization and architecture planning</li> </ul> <p><b>Skills</b></p> <p>The student can:</p> <ul style="list-style-type: none"> <li>• Test and debug API's communicating bi-directionally with web clients</li> <li>• Implement and scale stateful server applications</li> <li>• Reflect on advantages and disadvantages of architectural decisions</li> <li>• Structure and debug event-based systems</li> </ul> <p><b>Competences</b></p> <p>The student can:</p> <ul style="list-style-type: none"> <li>• Conduct technical decision-making and systems design for web development</li> <li>• Integrate heterogeneous components in web-based systems</li> </ul>
<p><u>Prerequisites for the exam, exam form and organization, evaluation criteria and any report structure and requirement</u></p> <p><b>The examination</b></p> <p>Oral exam is based on written project submission. The student submits a link to a version control repository with source code and system documentation. The submission must include a client and server application built using the tech stack used throughout the course. The exam duration is a total of 20 minutes including grading.</p> <p><b>Assessment</b></p> <p>7-point grading scale. The evaluation is an overall assessment based on written and oral performance.</p>

<b>Mobile Development</b>
<p>Teacher:</p> <p>Language: English</p> <p>This elective takes place by attendance</p>
<p><u>Contents:</u></p> <p>In this course the student will learn how to make cross-platform mobile (smartphone) apps. Main focus is programming, structure and state management. User experience will also be taken into consideration.</p> <p>10 ECTS</p>
<p><u>Learning objectives</u></p> <p><b>Knowledge</b></p> <p>The student must have knowledge of:</p> <ul style="list-style-type: none"> <li>• Core concepts of Dart programming language</li> <li>• Building UI with widgets in Flutter</li> <li>• Stack based navigation</li> <li>• Principles for separating logic and presentation</li> <li>• Reactive programming with BloC pattern</li> </ul> <p><b>Skills</b></p> <p>The student can:</p> <ul style="list-style-type: none"> <li>• Design user interfaces suitable for smartphones</li> <li>• Utilize phone features such as location, camera and notifications</li> <li>• Manage application state</li> <li>• Communicate with external APIs</li> <li>• Test and debug apps</li> <li>• Use packages and plugins</li> </ul> <p><b>Competences</b></p> <p>The student can:</p> <ul style="list-style-type: none"> <li>• Develop useful apps suited for smartphones</li> <li>• Reflect on design, structure and architecture of apps</li> </ul>
<p><u>Prerequisites for the exam, exam form and organization, evaluation criteria and any report structure and requirement</u></p> <p><b>The examination</b></p> <p>Exam is an individual oral presentation of self-made mobile application. The presentation is based on a written synopsis. The exam duration is 15 minutes including grading, wherein the student is expected to present for 5 minutes.</p> <p><b>Prerequisites for the exam:</b></p>

A synopsis for the project must be submitted along with a link to the code. The synopsis must have the full name of all contributors and a link to the source code on the front page.

**Assessment:**

7-point grading scale. The evaluation is based on the oral performance.

## Applications of Artificial Intelligence

Teacher: Henrik Boulund Meng Hansen

Language: English

This elective takes place by attendance

### Contents:

In this course the students will learn how to incorporate AI into their applications using Large Language Models (LLM) and how to orchestrate all crucial components in a proper manner. The students will also learn about LLMs and their parameters so they can determine the best LLM(s) for their concrete hardware and needs.

### Learning objectives:

#### **Knowledge**

The student must have knowledge of:

- What a large language model (LLM) is.
- Which providers there are on the market.
- Parameters stating the size and heaviness setting requirements for the hardware processing them.
- Where to find existing open-source models and compare them.
- Which systems do we have that can process open-source models.
- Relevant knowledge of how the gained skills and competences can be used in companies in the industry.

#### **Skills**

The student can:

- Develop basic applications using Python.
- Structure an LLM powered application in such a way that external data can be used for further improvement.
- Use proper Python libraries and tools for LLM communication.
- Use basic prompting techniques making LLM responses more concise and have control of the response.
- Use proper frameworks for designing user interfaces that makes the AI power be as simple as possible for the user.
- Use tools for testing core components and asserting LLMs and providers in terms of responses.
- Incorporate memory functionalities.
- Use good practice of designing, curating, and managing the context fed into Large Language Models.

#### **Competencies**

The student can:

- Develop applications that uses the power of LLMs.
- Determine which LLMs can be used for a specific hardware configuration.

Prerequisites for the exam, exam form and organization, evaluation criteria and any report structure and requirement

**The examination:**

Oral exam based on project and written submission. The exam duration is 15 minutes (the student presents 5 minutes, Q'n'A 5 minutes, assessment 5 minutes).

**Formal requirements to the project:**

A Python application that is empowered by LLMs using orchestrations tools and quality measures learned in class.

**Formal requirements to the written submission:**

A mini report at a maximum of 10 pages.

**Assessment:**

7-point grading scale.

The evaluation is an overall assessment based on the project, the report and the oral performance.