Bring ideas to life
VIA University College

Date: August 2024

Curriculum

Programme section

Bachelor of Engineering in Software Technology Engineering

Applicable to students enrolled in August 2024 and later.

However, students enrolled in summer 2023 will follow the study plan in Appendix 1.

For students enrolled before summer 2023, please refer to the 2022.

CONTENTS

Introduction

In accordance with the Executive Order on Bachelor of Engineering, the purpose of Bachelor of Engineering is to qualify the students to carry out the following professional functions nationally and internationally:

- Translate technical research results as well as scientific and technical knowledge into practical use in development tasks and in solving technical problems.
- Critically acquire new knowledge within relevant engineering areas.
- Independently solve common engineering tasks.
- Plan, implement and manage technical and technological facilities, including being able to involve societal, economic, environmental and occupational health and safety consequences in the solution of technical problems.
- Participate in collaborative and managerial functions and contexts at a qualified level with people who have different educational, linguistic and cultural backgrounds.

In addition, the education must qualify students to participate in further studies.

VIA's engineering programmes work on the basis of a common graduate profile. The graduate profile is a common profile for all VIA Engineers. The graduate profile is to be combined with the identity of the specific engineering programme.

At VIA Engineering, we are practice-oriented, project-oriented and world-focused. This is actualised in the form of qualified new graduates obtained through targeted teaching, relevant research and development, as well as collaboration and ongoing dialogue with the business community. The programmes must qualify graduates to handle practical and development-oriented business functions.

Programmes in English as well as admission of international students are hallmarks of our engineering programmes. This profile creates a unique opportunity to educate students who can act in a Danish context in an increasingly global market. Our lecturers have vast and solid practical experience and know how to anchor theory in practice through lab work, company visits and projects for and in collaboration with companies.

1 Identity of the programme

The Software Technology Engineering at VIA in Horsens is a study programme at the bachelor level. It is an applied engineering degree giving students skills and competences to be employed as software engineers after graduation. Software Technology Engineering graduates are qualified to:

- Apply research, theory, tools and methods from software engineering and natural science to conceive, design and implement solutions to practical engineering problems.
- Critically acquire new knowledge within the field of software engineering.
- Consider the social, financial, and environmental consequences of the suggested solutions.
- Work independently as well as in teams with members from different educational and cultural backgrounds.

The goals of the programme are achieved primarily by:

- Project work being an essential aspect of the teaching, where the academic elements of the programme are integrated via problem solving into a whole, with a focus on application-oriented and practical engineering work. In project work, emphasis is also placed on the students developing academic, professional, methodological, communicative, and personal skills.
- Collaborating with research environments and businesses in connection with the implementation of the teaching.
- Offering an international study environment, where parts of the study can be completed abroad, and where several courses are taught in English for Danish and foreign students alike.
- Actively using the student's engineering internship to bring about the exchange of knowledge and experiences between VIA and the profession.
- Achieving application- and practice-oriented skills primarily by utilising VIA's facilities within laboratories, manufacturing workshops and libraries, as well as completing internships and workshops.
- Priority being given to interdisciplinary focus areas within Digitalisation, Sustainability and Innovation and Entrepreneurship in the programme across the various semesters.

2 Graduate profile for VIA Engineers

Purpose
The newly graduated VIA engineer works problem-oriented, project- and team-based and contributes to advising, developing, inventing and quality-assuring products and solutions. The VIA engineer creates innovative, digital, sustainable and workable solutions to and for current and future societal and engineering challenges worldwide.

Skills
VIA Engineering educate holistic-thinking engineers who, through societal insight and personal development, can exploit the full potential of technology. Therefore, the skills of the VIA engineer range from highly specialised engineering skills to personal skills and the skills of the outside world.

Professional engineering skills
- Masters and applies – with critical reflection – highly specialised engineering knowledge.
- Works challenge-driven, innovative, and problem-oriented when developing engineering results.
- Integrates engineering and scientific knowledge, skills, and methods in solving engineering challenges.
- Designs, plans, simulates, manages, implements, and evaluates engineering solutions and products using digital and technological tools.
- Implements and operates solutions that match engineering needs within the industry.

Organisational skills
- Organises and manages projects and processes based on both risk assessment and market and business understanding.
- Collaborates inter-professionally with a global view and respect for the organisation, culture and methods of businesses and stakeholders.
- Involves knowledge of sustainability and circular economy in the development and implementation of new solutions.

Personal skills
- Works consistently with a curious and innovative mindset and seeks out, critically acquires and brings new knowledge into play throughout life.
- Communicates effectively and collaborates professionally with colleagues and people of different educational and cultural backgrounds.

3 Teaching and working methods

The engineering programme's priority focus areas within Digitalisation, Sustainability and Innovation and Entrepreneurship are integrated into relevant courses, so that together they constitute learning streams for all three areas.

Active and practice-oriented learning is supported by:

- Dialogue-based teaching with a high degree of active participation from students.
- Lectures in subjects where there is a large proportion of knowledge transfer. Lectures are usually combined with practice sessions with a student instructor or a lecturer.
- Project work and problem-oriented learning (PBL) are an essential part of the teaching, as the academic elements of the education programme are integrated into application-oriented engineering projects with emphasis on methodological problem solving.
- Projects being carried out in groups within the programme and in an interdisciplinary collaboration with other engineering programmes.
- Collaborating with research environments and businesses in connection with the implementation of the teaching.
- Offering an international study environment, where parts of the study can be completed abroad and where several courses are held in English for Danish and foreign students alike.
- The student's engineering internship being actively used to bring about the exchange of knowledge and experiences between VIA and the profession.

Application- and practice-oriented skills are primarily achieved by utilising VIA's facilities within laboratories, manufacturing workshops and library.

Teaching can be physical, online or located at another campus.

4 Structure and content

The programme is organised as an ordinary full-time higher education programme.
The structure and progression are stated in the overview on page 7.

The official duration of the programme is 3½ years, divided into 7 semesters of 30 ECTS, corresponding to 210 ECTS points in total.

The scope of each course or project is documented in the form of ECTS points (European Credit Transfer System). 1 ECTS point corresponds to a workload of approx. 27.5 hours for a student, an academic year of 60 ects thus corresponds to 1,650 hours of work for the student.

New students are admitted in August/September every year.

The study includes:

- Compulsory courses and projects
- Elective courses
- Internship
- Bachelor project
- Workshops

A semester consists of 3-4 courses, which are delimited courses. A course's scope can range from 5 to 10 ECTS points, and a project's scope from 10 to 15 ECTS points.

The purpose, scope, learning objectives and exams of courses are described in this curriculum. For a detailed and complete description of the individual courses, please refer to the course descriptions in force at any given time, which are available on VIA's website and on MyVIA.

The programme is structured as illustrated below:

Semester *Theme*	Course 5 ECTS	Course 5 ECTS	Course 5 ECTS	Course/ Project 5 ECTS	Project 5 ECTS	Project 5 ECTS
7 semester ***Electives***	**Elective Course**	**Elective Course**	**Elective Course**	**BPR2** Bachelor Project		
6 semester ***Innovation and Electives***	**Elective Course**	**Elective Course**	**Elective Course**	**BPR1** Bachelor Project Preparation	**IDE1** Semester Project Innovation and Entrepreneurship	
5 semester ***Internship***	**INP1** Internship					
4 semester ***Internet-of-Things***	**MAL1** Introduction to Machine Learning and AI	**ESW1** Embedded Software	**WEB2** Web Development 2	**DOC1** DevOps and Cloud	**SEP4** Semester Project	
3 semester ***Heterogeneous Systems***	**PRO3** Programming 3	**CAO1** Computer Architecture and Organisation	**DNP1** .NET Programming	**ADS1** Algorithms and Data Structures	**SEP3** Semester Project	
2 semester ***Client/Server Systems***	**PRO2** Programming 2		**SWE1** Software Engineering	**DBS1** Database Systems	**SEP2** Semester Project	
1 semester ***Single User Systems***	**PRO1** Programming 1		**WEB1** Web Development 1	**MSE1** Mathematics for Software Engineering	**SEP1** Semester Project	

6th and 7th semester, including electives and available specializations, are described in section 7.

5 Compulsory elements of the programme, 1st – 4th semester

All courses on the first 4 semesters are compulsory, and they all include a semester project. The overall goal of the semester projects is to connect the courses and for students to apply the skills acquired during the semester. Project methods, teamwork, communication, and documentation skills are taught in the context of the semester projects.

Each semester has a theme. The themes of the first four semesters are:

- 1st semester: Single User Systems
- 2nd semester: Client/Server Systems
- 3rd semester: Heterogeneous Systems
- 4th semester: Internet of Things

5.1 1st semester: Single User Systems

The goal of the 1st semester is to give the student knowledge and practical skills in object-oriented programming and systems development. The student will also learn about responsive web design, discrete mathematics, and algorithms. The semester is organised with a number of introductory courses and a semester project in which the students will design, implement, and document a single-user software system.

The scope of the semester is 30 ECTS.

The purpose of the courses, ECTS and assessment:

Programming 1 (PRO1) – 10 ECTS	**Exam prerequisites**	**Assessment**
The main purpose of the course is to provide students with the qualifications needed to understand the core object-oriented concepts and to implement smaller programs in Java from UML class diagrams.	1. Attendance (≥ 75%) 2. Participation in oral mid-term test 3. Completed or achieved credit transfer for WS1, Programming Workshop 1 If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual oral exam where the student will pick an unfamiliar programming exercise at random. The student must explain the UML involved and demonstrate how to perform the programming task using a laptop. The time allotted for the exam is 30 minutes including assessment. External assessment based on the Danish 7-point scale. Tools allowed: All. Re-exam: Conducted as the ordinary exam. Two re-exams (second and third attempt) will be scheduled in the beginning of second semester.
Web Development 1 (WEB1) – 5 ECTS		
The purpose of this course is to introduce a set of theories and tools in order for students to obtain a proficient level of knowledge and gain a practical set of skills for designing and developing responsive web sites for both desktops and mobile devices using basic web programming.	1. Attendance (≥ 75%) 2. Three course assignments handed in before deadline and approved. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Digital written exam duration of 2 hours (2 parts): Part 1: Multiple choice questions 30 minutes, weighing 25% Part 2: Short answer questions 90 minutes (explaining and writing code), weighing 75% Internal assessment based on the Danish 7-point scale. Tools allowed: Part 1: without aids Part 2: all aids allowed - including internet connection. Any types of communication between students or between a student and an external party is prohibited

		and will be considered a violation of the exam rules. Also, the use of AI tools, such as AI tools, ChatGPT or similar AI and Machine Learning tools and chatbots is not allowed. Re-exam: Re-exams may be oral.
Mathematics for Software Engineering (MSE1) – 5 ECTS		
The purpose of the course is to give the students the mathematical prerequisites to work with technical IT and specifically software engineering. The focus of the course will be to • supply competences in analysing and generalizing algorithms and problems that occur in the context of software development. • supply skills in expressing one's knowledge clearly and concisely. • formalize statements in a logically and computationally correct manner. • supply analytical problem-solving skills. After having successfully completed the course, the students will have acquired a solid understanding of the mathematics used in software engineering, a clear analytical mindset, as well as skills in the methodology of software engineering.	1. Attendance (≥ 75%) 2. Eight out of ten mandatory assignments approved. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: The exam has two parts: • The first part is a Flowlock exam in WISEflow. • The second part is a WISEflow exam without Flowlock. The exam has a total duration of 4 hours. The student will not be able to access the second part before the first part is concluded. Each part has a duration of 2 hours and weighs equally in the final grade. Internal assessment based on the Danish 7-point scale. <u>Tools allowed:</u> In the first part the students are allowed to use any notes, books, and/or other written/printed material and will have access to pdf files on their laptop. The student may bring their own calculator. In the second part all supplementary materials and aids are allowed, e.g., using a computer to do calculations, but the student is not allowed to go online, i.e. no Internet access. Even more so, it is not allowed to use AI-tools such as CoPilot, ChatGPT, Bing, etc. as per the general VIA rules. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Re-exam: Re-exams may be oral.
Semester Project (SEP1) – 10 ECTS		
The purpose is to develop and document a single user system and a basic introduction to team-based project work.	None	Type of exam: Group exam with individual assessment. Exam is based on the project report and process report, which must be handed in before deadline and in accordance with VIA Engineering guidelines including guidelines on Formalities. Group presentation of 5 minutes times the number of group members followed by a joint exam of 15 minutes times the number of group members including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the exam. Internal assessment based on the Danish 7-point scale. Tools allowed: All.

		Re-exams: Students who failed a semester-project must make a new project. Students who fail a semester project must attend an information meeting at the end of the summer exam period. At this meeting, students will be notified about the process of the re-exam and students will form groups. Furthermore, specific deadlines and exam dates are announced at this meeting. The project must be handed in before the stipulated deadline. An exam schedule will be posted after the hand in deadline. Exam will take place as the ordinary exam.

The learning objectives of the courses (knowledge, skills, and competencies) can be found in Appendix 1.

5.2 2nd semester: Client/Server Systems

During the 2nd semester, the student will learn about software engineering methods, database development and system development methods, and the student will accumulate the programming skills necessary to build client/server systems. The semester is organised as a number of compulsory courses, along with a semester project in which the students will develop a client/server system.

The scope of the semester is 30 ECTS.

The purpose of the courses, ECTS and assessment:

Programming 2 (PRO2) – 10 ECTS	**Exam prerequisites**	**Assessment**
The purpose is to qualify the student to understand and master the concepts and techniques of object-oriented system development and programming, including Client/Server programming and basic understanding of computer networks.	1. Attendance (≥ 75%) 2. Three or four course assignments handed in before deadline. 3. Completed or achieved credit transfer for WS2, Programming Workshop 2 If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual oral exam, 20 minutes, without preparation. • The student will draw from a pool of previously known questions. • The student will explain concepts and theories from the course, preferably using the course work as reference. External assessment based on the Danish 7-point scale. Tools allowed: All Re-exam: Conducted as the ordinary exam.
Database Systems (DBS1) – 5 ECTS		
The course has two purposes. Firstly, students are to learn methods for designing, implementing and operating single-user relational databases. Secondly, students are to learn the main principles, architecture and technologies of a typical relational database management system (RDBMS).	1. Attendance (≥ 75%) 2. Three course assignments handed in before deadline. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: The students must import a database and dataset before the exam. The first part of the exam will ask questions about this dataset. This dataset will be made available through WISEflow several days before the exam. Digital written exam (2 parts), 4 hours: Part 1: Multiple choice and written answers in WISEflow Part 2: Design and implementation External assessment based on the Danish 7-point scale. Tools allowed: All aids are allowed included access to online material. However, it is not allowed to use AI tools such as ChatGPT and similar AI and Machine

		Learning driven tools and chatbots. Re-exam: Conducted as the ordinary exam.
Software Engineering (SWE1) – 5 ECTS		
The purpose is to qualify the student to apply software engineering concepts used to develop object-oriented software. Structure the software development process by applying SCRUM and Unified Process to conduct Analyse, Design and Test-descriptions to exemplify a final solution from a real-life problem. This involves requirement capturing (Use Cases and non-functional requirements), analysis, domain models, interaction diagrams, design classes, design patterns and test-descriptions etc.	1. Attendance (≥ 75%) If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual oral exam, 20 minutes. Exam is without preparation and based on a drawn question. The questions will be known before the exam. External assessment based on the Danish 7-point scale. Tools allowed: All Re-exam: Conducted as the ordinary exam. .
Semester Project (SEP2) – 10 ECTS		
The purpose is to develop and document a client/server system. The purpose of the PBL part of the course is to promote the students' competencies in collaboration, planning and problem analysis.	None	Type of exam: Group exam with individual assessment. Exam is based on the project report and process report, which must be submitted before deadline and apply with the formalities criteria stated under the Software Engineering specific Guidelines. Group presentation approx. 20 minutes followed by a joint exam with a joint discussion and individual question rounds for approx. 20 minutes per student including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the exam. Internal assessment based on the Danish 7-point scale. Tools allowed: All. Re-exams: Students who failed a semester must make a new project. Students who fail a semester project must attend an information meeting at the end of the summer exam period. At this meeting, students will be notified about the process of the re-exam and students will form groups. Deadline for hand in of the project is mid-August (exact date will be informed at the meeting). There will be no guidance in the period up to hand in. Oral assessment of the project takes place before the start of the autumn semester.

The learning objectives of the courses (knowledge, skills and competencies) can be found in Appendix 1.

5.3 3rd semester: Heterogeneous Systems

The aim of the 3rd semester is to design and implement heterogeneous software solutions including the use of the programming languages Assembler, C, Java and C#. The semester is built around a larger semester project in which students will integrate several programming languages.

The scope of the semester is 30 ECTS.

The purpose of the courses, ECTS and assessment:

Programming 3 (PRO3) – 5 ECTS	**Exam prerequisites**	**Assessment**
The students should be introduced to basic theory of distributed systems and security technology and be able to design and implement a secure distributed system.	1. Attendance (≥ 75%) 2. Completed or achieved credit transfer for WS3, Computer Networking Workshop. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: • Individual oral exam, 20 minutes, without preparation based upon course work • The student will draw from a pool of previously known questions. • The student will explain concepts and theories from the course, using the course work as reference • The student will start with a prepared presentation External assessment based on the Danish 7-point scale. Tools allowed: All Re-exam: Conducted as the ordinary exam.
Computer Architecture and Organisation (CAO1) – 5 ECTS		
The main purpose of the course is to gain a basic understanding of the organisation and design of computers and how a computer works. Focus will be on the central processing unit (CPU) and the necessary logic involved in building a CPU.	1. Attendance (≥ 75%) 2. Two assignments approved. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Written exam, 2 hours The exam is digital, and it is the responsibility of the student to bring a computer that works together with WISEflow and FLOWlock. External assessment based on the Danish 7-point scale. Tools allowed: Course literature and personal notes. Re-exam: Conducted as the ordinary exam.
.NET Programming (DNP1) – 5 ECTS		
The purpose is to qualify the student to describe and implement the basic concepts of the C# programming language and the .NET developer platform with a focus on ASP.NET.	1. Attendance (≥ 75%) 2. Course assignment handed in before deadline. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual written exam, 4 hours, consisting of programming exercises. Internal assessment based on the Danish 7-point scale. Tools allowed: All aids are allowed included access to online material. However, it is not allowed to use AI tools such as ChatGPT and similar AI and Machine Learning driven tools and chatbots. Re-exam: Re-exams may be oral.

Algorithms and Data Structures (ADS1) – 5 ECTS		
The purpose is to qualify the student to design, implement and analyse different algorithms and to become acquainted with different advanced data structures	1. Attendance (≥ 75%) 2. Three course assignments handed in before deadline. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual written exam, 3 hours. Internal assessment based on the Danish 7-point scale. Tools allowed: - Course literature according to the course description. - Personal notes. Re-exam: Re-exams may be oral.
Semester Project (SEP3) – 10 ECTS		
The purpose is to • develop and document a distributed system • herein account for the security aspects of the system. The purpose of the PBL part of the course is to promote the students' independent knowledge application, critical thinking and holistic understanding.	None	Type of exam: Oral group exam with individual assessment. Exam is based on project and process report, which must be submitted before deadline and apply with the formalities criteria stated under the Software Engineering specific Guidelines. Group presentation approx. 20 minutes followed by joint exam with joint discussion and individual question and answer sessions for approx. 20 minutes per student including evaluation. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual's performance during the exam. Internal assessment based on the Danish 7-point scale. Tools allowed: All. Re-exam: Students who failed a semester must make a new project. Students who fail a semester project must attend an information meeting at the end of the summer exam period. At this meeting, students will be notified about the process of the re-exam and students will form groups. Deadline for hand in of the project is mid-August (exact date will be informed at the meeting). There will be no guidance in the period up to hand in. Oral assessment of the project takes place before the start of the autumn semester.

The learning objectives of the courses (knowledge, skills and competencies) can be found in Appendix 1.

5.4 4th semester: Internet of Things

The 4th semester brings it all together and students will conceive, design and implement a software solution including hardware sensors, a web-based user interface and a persistent multiuser backend infrastructure. The solution must contain self-constructed electronics, and make use of the Java, C#, C and assembler programming languages.

The scope of the semester is 30 ECTS.

The purpose of the courses, ECTS and assessment:

Introduction to Machine Learning and AI (MAL1) – 5 ECTS	**Exam prerequisites**	**Assessment**
In this course, students will acquire both theoretical knowledge and practical skills in the application of machine learning methodologies to a spectrum of data types, encompassing both structured and unstructured datasets. The curriculum is designed to ensure that participants thoroughly understand and can adeptly utilize advanced tools and techniques essential for data preparation, preprocessing, and exploration. Students will be equipped to discern underlying structures and make informed predictions. Central to the course are four primary topics: - Classification: Understanding and categorizing data into predefined classes. - Regression: Predicting continuous outputs based on data input. - Clustering: Identifying the inherent groupings within datasets. Dimensionality Reduction: Simplifying complex data structures without losing critical information.	1. At the end of the course, the student must upload a 1-page summary of each of their 6 assignments as well as a 2-page summary of their group project. The summaries must include a brief description of: a. the assignment problem b. how the assignment was solved, e.g., data acquisition, data preparation, feature engineering, feature extraction, etc. c. the algorithms that were used to solve the problem. d. the performance of the final model e. a reflection of the learning outcome of solving the assignment. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: The exam is a 20-minute oral exam that departs from one of the six assignments that the student made during the semester. The exam will also include an examination of the group project report. The final grade will be based on an overall assessment of the six assignments, the group project report, and the oral exam. External assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam.
WEB Development 2 (WEB2) – 5 ECTS		
The purpose of the course is to introduce the students to modern web development, by going in depth with the JavaScript language & the principles behind prominent JavaScript frameworks.	1. Attendance (≥ 75%) 2. Two course assignments approved. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual written exam, 1 hour Internal assessment based on the Danish 7-point scale. Tools allowed: None. Any type of communication between students or between a student and an external party is prohibited and will be considered a violation of the exam rules.

		Re-exam: Re-exams may be oral
Embedded Software (ESW1) – 5 ECTS		
The purpose is to qualify the student to apply basic concepts in embedded programming using the C-programming language on embedded micro-controllers and using interfaces (APIs) for different sensors and actuators.	1. Attendance (≥ 75%) If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual written exam, 3 hours. Internal assessment based on the Danish 7-point scale. Tools allowed: Course literature according to the course description Personal notes on paper Access to local pdf-files Laptop (no access to general internet) Re-exam: Same as the ordinary exam
DevOps & Cloud (DOC1) – 5 ECTS		
The purpose of the course is to impart the student with a general understanding of how software development activities are coupled to the other parts of the value chain, through the introduction of a contemporary set of ideas, tools and concepts (known as DevOps) that help deliver software in a fast and reliable manner. The student will learn how to use this knowledge to create and maintain a consistent toolchain that supports execution of a software project by automating workflows and integrating the processes of software development and IT operations.	1. Attendance (≥ 75%) If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual oral exam, 20 minutes. Internal assessment based on the Danish 7-point scale. Tools allowed: All Re-exam: Conducted as the ordinary exam.
Semester Project (SEP4) – 10 ECTS		
The main purpose is to conceive, analyse, design and implement an Internet of Things (IOT) software solution in larger groups with a focus on continuous integration and delivery. The software solution will contain custom IoT-hardware and drivers to retrieve sensor data, a cloud backend infrastructure and a web-based user interface for facilitating sensor data visualization and user interactions with the IoT-hardware.	None	Type of exam: Oral group exam with individual assessment. Exam is based on project report, process report, source code and group video presentation, all of which must be handed in before deadline. Group exam of 3-4 students from the group at a time with joint discussions and individual questions for 15 minutes per student including grading. Internal assessment based on the Danish 7-point scale. Tools allowed: All. Re-exam: Students who fail a semester project must attend an information meeting at the end of the summer exam period.

The purpose of the PBL part of the course is to promote the students' competencies in cross-professional collaboration.		At this meeting, students will be notified about the process of the re-exam and students will form groups, if possible, in relation to the number of failed students. Based on the feedback the students have received after the ordinary exam, they must prepare a new project, or the failed project must be improved. Deadline for hand in of the project is mid-August (exact date will be informed at the meeting). There will be no guidance in the period up to hand in. Oral assessment of the project takes place before the start of the autumn semester.

The learning objectives of the courses (knowledge, skills and competencies) can be found in Appendix 1.

6 Internship, 5th semester

IT-INP1, passed/not passed, internal assessment

The internship comprises a semester of 30 ECTS and is timewise placed in the 5th semester of the programme. As a general rule, the internship period is paid and settled in a private or public company in Denmark or abroad. Student must be on an internship for a minimum of 20 full weeks excluding holidays, etc.

The purpose of the internship is for the student to acquire insight into practical common engineering work corresponding to engineering assistant work, combined with the integrated application of the acquired concepts, methods and techniques of the discipline in the first four semesters.

The student will find an internship, which must be approved by VIA, who appoints a supervisor for the intern. In collaboration with the company, the student prepares a plan for the internship with appertaining formulated assignments.

The basis for assessment of the internship is an ongoing report from the student to VIA, feedback from the internship company and a presentation where the supervisor can ask elaborating questions about the content of the internship.

If the engineering internship is interrupted before the end of the agreed internship period, the internship supervisor must, in consultation with the head of the education programme, assess whether the internship has been of sufficient length and content for there to be grounds for passing the internship present.

7 6th and 7th semester

The 6th and 7th semester consist of one mandatory course (BPR1), elective courses, and projects.

The Software Engineering degree can be completed without specialisation, or the student can choose one of these specialisations:
- Interactive Media
- Data Engineering

If too few students have chosen one of the specialisations, VIA reserves the right to cancel the specialisation in question.

In order to obtain a specialisation, the student must choose three electives (15 ECTS) within the subject area of the specialisation in question, and a significant part of the bachelor project must be within the same specialisation's subject area.

An overview of elective course offered by Software Engineering is shown below in section 8.

It is also possible to choose one elective course (5 ECTS) other than what is offered in the programme's elective survey. Selecting a course from other programmes must be pre-approved by an Engineering study counsellor.

7.1 6th semester

A compulsory 10 ECTS semester project (IDE1) within the area of innovation and entrepreneurship is completed in project groups of students from different engineering programs at VIA.
In addition to IDE1, the student must pass BPR1 and three electives.

Bachelor Project Preparation (BPR1) – 5 ECTS	**Exam prerequisites**	**Assessment**
The purpose of this course is to thoroughly prepare students for their Bachelor Project. This preparation is twofold: it involves both practical skills in project preparation and planning as well as an understanding of the underlying scientific setting that informs these practices. Students will engage in critical analysis of significant problem sets within their professional area, exploring alternative solutions and understanding the demands of companies and their environments. A key addition to this course is an exploration of the philosophy of science as it applies to engineering. This will equip students with a deeper understanding of the scientific principles and theories that underpin their field, enabling them to apply this knowledge in new and dynamic situations. In terms of project preparation, students will form groups, analysing and documenting their group composition in writing. They will establish a set of rules for effective collaboration within these groups. These activities will be guided by interpersonal competencies and will involve a reflection on principles included in their previous projects. Furthermore, students will learn to articulate, in writing, a proposed problem for study. This will involve developing a project description that incorporates scientific knowledge and work methods relevant to their field. The course also includes an analysis phase where students establish software requirements for their proposed project. Each bachelor project group should consist of 2-3 students. Students are encouraged to find their projects in association with an external company or organization, fostering a connection between academic learning and real-world application.	To qualify for the exam, the student must have their group project description approved. The hand-in for the project description lies after Element 2 but before Element 3 as described below. In case the project description is not approved, the student cannot attend Element 3, and a date for resubmission will be announced. This date will lie before the reexam period.	The course is assessed continuously through three elements: Element 1 (34%): A 30-minutes individual multiple choice test within the topics of philosophy of engineering. This test lies during the semester. Element 2 (33%): A small group assignment (max 5 pages) that includes a group contract. The group contract is not included in the max 5 pages. The hand-in of the assignment lies during the semester but after the test in Element 1. Element 3 (33%): A 30-minutes individual multiple choice test within the project-related topics. The test lies during the exam period. The student will receive one final grade based on these three elements and each element is weighted as described. Internal assessment based on the Danish 7-point scale. Re-exam: Should the student receive a non-passing final grade, they must redo the Elements in which they did not receive a passing grade. These reexams are held during the re-exam period.

Innovation and Entrepreneurship project (IDE1) – 10 ECTS	Exam prerequisites	Assessment
A cross-sectoral semester project that aims to develop and document an across disciplinary innovation and entrepreneurship project based on primary data collection.	Hand in 6 written assignments in English to be approved in WISEflow before deadline.	Type of exam: Exam is based upon the IDE1-report submitted in WISEflow before deadline. The group presents their prototype/pretotype. The exam room can be customized by the group to support the presentation. The exam is in English. Group exam with individual assessment. Group presentation approx. 15 minutes followed by joint evaluation with joint discussion and individual question rounds for approx. 60 minutes per group including assessment. Individual grades are given based on an overall assessment of the submitted work as well as the individual's presentation during the exam. External assessment based on the Danish 7-point scale. Tools allowed: All. Re-exam: Based on the submitted project, the examiner gives the student guidance on necessary improvements in relation to passing the exam (possibly, that a new project should be prepared). The students are informed about specific deadlines and details of the project work. Project groups are formed if possible. No further guidance is provided in the period leading up to submission. The project is assessed at an oral project exam.
Electives	**Exam prerequisites**	**Assessment**
For further details see section 8		

Primarily for international students admitted for a shorter period than one semester, SEP6 is offered in a modified version called SPRAU.

Project (SPRAU) 10 ECTS (max)	Exam prerequisites	Assessment
The purpose of the Project is to evolve the student's ability to solve a relevant Software Engineering problem and document the solution. The more specific problems chosen to be dealt with shall be connected to the student's background from her/his home university. In a group, students must be able to analyse, design, implement and test complex problems and be able to carry out well-documented and tested solutions.	None	Type of exam: Group exam with individual assessment. Exam is based on the project report(s), which must be submitted before deadline and apply with the formalities criteria stated under the Software Engineering specific Guidelines. Group presentation approx. 5-10 minutes per student, followed by a joint exam with a joint discussion and individual question rounds for approx. 25 minutes per student including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the exam. Internal examiner.

7.2 7th semester

At the 7th semester the students must complete additionally three elective courses as well as the bachelor project (BPR2) which constitutes 15 ECTS credits of the programme. The bachelor project is described further below and in section 10.

Electives	**Exam prerequisites**	**Assessment**
For further details see section 8		
Bachelor Project (BPR2) – 15 ECTS	**Exam prerequisites**	**Assessment**
The project should be based on a software engineering problem, with a project description created in the BPR1 course. The purpose of the bachelor project is to provide students with practical experience in addressing complex, real-world problems related to software engineering, mirroring the types of challenges they may face in their professional careers. The BPR2 course allows students to explore a wide range of topics - from developing specific software solutions at the behest of a partnering company to undertaking broader investigative projects that may not necessarily result in extensive code production but instead focus on exploring potential solutions to industry-relevant issues. Students are encouraged to identify a problem or area of interest, apply relevant theories and methodologies, and work collaboratively, often with the option to partner with external organizations. The project is designed to foster not only technical skills but also competencies in problem-solving, project management, and team collaboration, ensuring that students gain a comprehensive understanding and hands-on experience that aligns closely with professional software engineering practices. The group must document not only the project results but also the process, incorporating theories related to teamwork, interpersonal competencies, and workflow. Each bachelor project group should consist of 2-3 students.	Passed all other courses of the bachelor programme.	Exam type: Group exam with individual assessment based on the reports which must be handed in on time and apply with the formalities criteria stated under the Software Engineering specific Guidelines. The exam consists of two parts: First, a group presentation of the project (20 minutes) Afterwards, a joint exam of 30 minutes per group member, including assessment. Individual grades are given based on an overall assessment of the submitted work as well as the individual´s performance during the exam. External assessment based on the 7-step scale. Tools allowed: N/A Re-exam: Based on the feedback the students have received after the ordinary exam, they must either prepare a new project, or the failed project must be improved. There is no supervisor attached for re-exam attempts.

8 Electives

On the Software Technology Engineering programme, the following electives are offered regardless of any specialisation:

Elective courses may be cancelled in case of an insufficient number of students registered for the course. In case of oversubscription for an elective course, registrations may be declined. Unless otherwise stated in the course description, selection of participants will be made by lot.

Electives (5 ECTS)	**Course purpose**	**Assessment** (All re-exams may be oral exams)
Programming Concepts and Languages (PCL1) Within the specialisation: None	The purpose of the course is to qualify the student to: - Understand various programming concepts, paradigms and get knowledge about how different paradigms appear in different programming languages - Get thorough knowledge about the functional programming paradigm - Apply different paradigms to specific problems in different languages	Exam prerequisites: None Type of exam: Individual written exam, 3 hours Internal assessment based on the Danish 7-point scale. Tools allowed: All aids are allowed included access to online material. However, it is not allowed to use AI tools such as ChatGPT and similar AI and Machine Learning driven tools and chatbots. Re-exam: Conducted as the ordinary exam.
Applied Linear Algebra (ALI1) Within the specialisation: None	The purpose of the course is to equip the student with basic knowledge about linear algebra and its applications. This will enable the student to not only understand but also apply linear algebra in solving practical engineering problems. Skills in linear algebra are of high importance when dealing with scientific computing, image processing graphics, robot technology, algorithmics, coding theory, and more. As an example, the founders of Google have cited their course in linear algebra as the backbone of Google's PageRank feature (i.e., ordering web pages after importance). The course familiarizes students with scalars, vectors, matrices, determinants, operations on vectors and matrices, and systems of linear equations in matrix form. The course also presents applications of matrix theory to linear models, including examples from engineering.	Exam prerequisites: None Type of exam: The final exam has two parts. • The first part is a Flowlock exam in WISEflow. • The second part is a WISEflow exam without Flowlock. The second part must be completed in the Jupyter Notebook environment and the answers must be submitted in WISEflow. The exam has a total duration of 4 hours. The student will not be able to access the second part before the first part is concluded. Each part has a duration of 2 hours and weighs equally in the final grade. Internal assessment based on the Danish 7-point scale. Tools allowed: In the first part the students are allowed to use any notes, books, and/or other written/printed material and will have access to pdf files on their laptop. In the second part all supplementary materials and aids are allowed, e.g., using a computer as a reference work. It is not allowed, however, to use AI-tools such as CoPilot, ChatGPT, Bing, etc. as per the general VIA rules. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Re-exam: Re-exams may be oral.
Compiler Construction (CMC1) Within the specialisation: None	The students should be able to design a simple programming language, and design and implement a compiler for the language.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes. The exam is based upon a subject found by draw, and the answers are based on the project produced in the course and handed in according to deadline. Internal assessment based on the Danish 7-point scale.

		Tools allowed: All Re-exam: Conducted as the ordinary exam
Domain Centric Architecture (DCA1) Within the specialisation: None	The purpose is to qualify the student to understand and master the concepts and techniques of software architecture, test driven development, implementations of both. The course will provide students with the qualifications needed to understand how to: • Discuss various architectural styles • Implement several architectural patterns • Apply a subset of strategic and tactical Domain Driven Design patterns • Drive software development with automated tests	Exam prerequisites: None Exam type: Individual oral exam, 20 minutes, without preparation. The exam is based upon course assignments, which must be submitted before the given deadline. The student will draw from a pool of known questions, and they will explain concepts and theories from the course, using the course work as reference. The student will start with a prepared presentation. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam.
IT Security and Cryptography in Practice (SCP1) Within the specialisation: None	This course is student-driven and is focused on the real-world application of IT security in a practical environment. It includes experiencing how information and risk, threats and attacks, cyber security architecture and operations, secure systems hardening, and usability and cyber security management are applied to provide resilience in practical context. Students who do this course will obtain practical experience in the design, implementation, and evaluation of cyber security approaches.	Exam prerequisites: 3 mandatory assignments handed in: 1) A 1-page summary of their project idea. 2) A 1-page summary of their midterm seminar report. 3) A 1-page summary of their final report. If a student fails to meet one or more of the above mandatory assignments, the student will be given an extra assignment, to qualify for re-exam. The scope of this assignment depends on the scope of the missing requirements. Type of exam: The exam has ongoing assessment. Midway exam based on Midway Paper (30%) Final Exam based on Final Paper (70%) Internal assessment based on the Danish 7-point scale. Tools allowed: All Re-exam: The re-exam consists of two parts: 1) A 1-page summary of each of the main topics in the course, incl. the student's own topic (10%) 2) A 20-minute oral examination based on Final Paper (90%) The student may choose to resubmit a revised version of the final paper. The main topics of the course are determined by the students at the beginning of the course and consists of the topics of their final paper.

Project Management (PRM1) Within the specialisation: None	The purpose of this course is for the students to be familiar with the tools that can help the project manager being successful in his or her work. Focus will be on how to organize a complex cross-disciplinary project and apply relevant tool to minimize the risk of failure.	Exam prerequisites: None Exam type: Written examination consisting of two reports. One being the written group report on the case work handed in before end of semester and one being a final individual reflection report produced after hand in of group report. Group report: Max 15 pages Individual report: Max 4 pages (Normal pages: 2400 characters including spaces, figures do not count as characters) Internal assessment based on overall assessment of written group report and individual report. Grading based on the Danish 7-point scale. Tools allowed: All Re-exam: Conducted as the ordinary exam.
Calculus (CAL1) Within the specialisation: None	In the course, the students attain knowledge about and practical experience in applying the methods and tools of calculus. Most importantly, the course will enable the student to apply differential and integral calculus in solving a wide range of problems.	Exam prerequisites: None Type of exam: Ongoing tests in the form of course assignments, (10%) and an oral exam (90%), 20 minutes, including everything. A selection of approximately 10 of the exercises from the course will form the basis for the exam. During the exam, the student will randomly draw and present one of these exercises. There is no preparation time. The exam will then evolve into a general discussion of the course curriculum. Internal assessment based on the Danish 7-point scale. Tools allowed: The student is allowed to bring their notes to the oral exam, but these must be placed on the table during the examination. During the presentation, the student is allowed to consult their notes if they need to, but excessive use of the notes will count negatively towards the grade. During the discussion that follows the presentation, the student is not allowed to consult their notes. Re-exam: Conducted as the ordinary exam.
Full-stack Development (WEB3)	The purpose of this code is to learn to design and implement full stack web applications.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, without preparation. Exam departs from 2 of the 6 course assignments, handed in according to deadline. Exam covers the assignments and the part of the syllabus relevant to the assignments. The final grade will be based on an overall assessment of the 6 assignments and the oral examination. Internal assessment based on the Danish 7-

		point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam.
User Experience and Usability (UXU1) Within the specialisation: Interactive Media	To introduce students to User Experience and Usability Understanding the user is an important part of design. In this course the student gets to dive into the mind and feelings of the user to gain skills in designing better solutions. Cognition, emotions, authenticity and hedonics are topics within this area. In this journey the key term experience is explored in particular within usability, and different kinds of usability tests are conducted e.g. using tools like eye tracking glasses and other neuro sensors.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, without preparation. Exam is based on both a theory question from the course syllabus and practical use documented in the course assignment, which must be handed in before deadline. Internal assessment based on the Danish 7-point scale. Tools allowed: all Re-exam: Same as the ordinary exam.
Game Development (GMD1) Within the specialisation: Interactive Media	The purpose of this course is to provide the student with knowledge, skills and competences to develop games and interactive experiences using a real-time game engine. The course provides a multidisciplinary perspective to game development but is primarily rooted in the role of the software engineer, focusing on code architecture in a script-based environment. Through weekly exercises and a project, the course covers various topics, which will provide the student with the qualifications needed to develop interactive experiences across various platforms including PC, mobile and interactive tables.	Exam prerequisites: None Type of exam: Written exam in the form of a course assignment, handed in before deadline. If the course assignment has been carried out as group work, it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each student must hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam (individually or group). Students who fail the ordinary exam will be given a new deadline to hand in.
Digital Multi Media (DIM1) Within the specialisation: Interactive Media	To introduce students to basic principles of each media type - text, graphics, audio, animation, and video - describing their digitization and progressing onto issues that arises when media are combined.	Exam prerequisites: None Type of exam: Written exam in the form of a course assignment, handed in before deadline. If the course assignment has been carried out as group work, it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each student must hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A

		Re-exam: Conducted as the ordinary exam (individually or group). Students who fail the ordinary exam will be given a new deadline to hand in.
XR Development (XRD1) Within the specialisation: Interactive Media	In this course students learn how to implement augmented- and virtual reality applications, reflect on their relevant use cases and gain an understanding of the underlying technology that enables the experiences.	Exam prerequisites: None Type of exam: Written assignment spanning the semester. Assessment is based on projects developed in groups where it must be clearly marked which parts of the applications each group member contributed to. Furthermore, each student must hand in an additional 1-2 pages (one page is considered 2400 characters) of individual reflections on the work they have done. Assessed by the lecturer and an internal assessor without oral presentation or defence. Grading based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam (individually or group assignment if possible). Students who fail the ordinary exam will be given a new deadline to hand in.
Digital Signal Processing (DSP1) Within the specialisation: None - but related to the field: Internet of Things	The purpose of the course is to equip the student with basic knowledge about the fundamentals of Digital Signal Processing and its applications. Starting from the basic definition of a discrete-time signal, we will work our way through sampling, filter design, and Fourier analysis to build a basic DSP toolset. Signal processing is one of the fundamental theories and techniques to construct modern information systems. For example, audio, speech, and image processing, computer graphics, biomedicine all apply digital signal processing. In fact, digital signal processing is used to develop algorithms that can diagnose heart disease and can even be used to detect hostile drones. The course familiarizes the student with digital signals, sampling theory, digital filtering, the Fast Fourier Transform, power spectrum, and feature extraction.	Exam prerequisites: None Exam type: Individual oral exam, 20 min. Exam is based upon an assignment handed in before deadline. The students will present the assignment in the form of a demonstration, followed by questions about the signal processing and feature extraction methods as well as the MATLAB programming. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam (new assignment).
Embedded Operating Systems (EOS1) Within the specialisation: None - but related to the field: Internet of Things	Students will acquire basic knowledge about the Linux Operating System and practical experience in development of an IoT-device based on a combination of self-made and Open-Source software.	Exam prerequisites: None. Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment based on the Danish 7-point scale. Tools allowed: Laptop Course hardware kit Re-exam: Conducted as the ordinary exam.

Real-Time Programming (RTP1) Within the specialisation: None - but related to the field: Internet of Things	The main purpose of the course is to provide students with the qualifications needed to understand central concepts and characteristics about embedded real-time programming.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment based on the Danish 7-point scale. Tools allowed: Laptop Re-exam: Conducted as the ordinary exam.
Hardware Oriented Programming (HWP1) Within the specialisation: None - but related to the field: Internet of Things	The purpose of the course is: - To provide the student with knowledge about the technical details of an industrial microcontroller used for embedded systems from a programmer's point of view. - To qualify the student to implement simple low-level drivers for various hardware devices. - To qualify the student to implement low-level software for an embedded system in C. - To qualify the student to do Unit testing of embedded C.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment based on the Danish 7-point scale. Tools allowed: Laptop Course hardware kit Re-exam: Conducted as the ordinary exam.
Data Analytics Infrastructure (DAI1) Within the specialisation: Data Engineering	The course introduces the student to selected topics in the design and implementation of infrastructure to support data analytics. Within this area, the course will introduce students to different tools and techniques for data acquisition, cleansing and integration. The students will also be introduced to data modelling for analytics and basic visualization.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes without preparation. Exam is based upon four course assignments handed in before deadline. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam
Machine Learning for Artificial Intelligence (MAL2) Within the specialisation: Data Engineering	This course explores the fundamental concepts, techniques, and applications of deep learning in the context of artificial intelligence (AI). This course is designed to provide students with a comprehensive understanding of how deep learning methods can be leveraged to solve complex AI problems.	Exam prerequisites: None. Exam type: Individual oral exam, 20 minutes without preparation. At the exam, the student will randomly draw one of the portfolio assignments. The exam will then take place as a discussion of this assignment, the students' group project and the curriculum in general. Internal assessment based on the Danish 7-point scale. Tools allowed: The student is expected to bring their portfolio assignments and their final project to the oral exam, such that they are able to display and run their code.

		Re-exam: Conducted as the ordinary exam
ERP Systems SAP ABAP/4 Programming (ERP1) Within the specialisation: Data Engineering	There are two main purposes of this course: The first purpose (36 lessons) is to introduce students to the aspects of analysis, design, coding, and testing company specified programs in the SAP ABAP environment. The second purpose (12 lessons) is to give the students the opportunity to work further with a selected part of the SAP System: - Object Oriented programming in ABAP - Development of SAP S/4 HANA - SAP for production planning etc.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes. The exam is in two parts: First part is a presentation and discussion of selected parts of the course work (which consists of 3 mandatory written course assignments, handed in before deadline). Second part is drawn question from the theory of the course. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam. New assignments are accepted.
Stochastic Modelling and Processes (SMP1) Within the specialisation: Data Engineering	The ubiquitous presence of uncertainty and noise in the engineering sciences makes it mandatory to understand and quantify random phenomena. To achieve this goal the course will provide a solid introduction to the theory of stochastic processes. Special attention is given to applications and the student will model and analyse complex stochastic situations as encountered in practice. The applications include examples from various engineering fields such as information technologies and communications, signal processing, and more.	Exam prerequisites: None The final exam has two parts. • The first part is a Flowlock exam in WISEflow. • The second part is a WISEflow exam without Flowlock. The second part must be completed in the Jupyter Notebook environment and the answers must be submitted in WISEflow. The exam has a total duration of 4 hours. The student will not be able to access the second part before the first part is concluded. Each part has a duration of 2 hours and weighs equally in the final grade. Internal assessment based on the Danish 7-point scale. Tools allowed: In the first part the students are allowed to use any notes, books, and/or other written/printed material and will have access to pdf files on their laptop. The student may bring their own calculator. In the second part all supplementary materials and aids are allowed, e.g., using a computer as a reference work. It is not allowed, however, to use AI-tools such as CoPilot, ChatGPT, Bing, etc. as per the general VIA rules. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Re-exam: Re-exams may be oral.
Business Intelligence (BUI1) Within the specialisation: Data Engineering	Business intelligence is the delivery of accurate, useful information to the appropriate decision makers within the necessary time frame to support effective decision making.	Exam prerequisites: None Type of exam: Oral exam based on the course assignment, which must be handed in before deadline.

	The main purpose of the course is to equip the student to work with realistic business data using professional business intelligence tools in order to develop analytical solutions for businesses.	Approximately 20 minutes incl. discussion of examinee's performance, without preparation. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam
No-SQL versus relational databases (NSQ1) Within the specialisation: Data Engineering	This course will provide students with knowledge of strengths and weaknesses of two fundamentally different approaches to database management systems.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes without preparation. Exam is based upon two course assignments handed in before deadline, and it is covering mandatory course work and theory covered in the course. Internal assessment based on the Danish 7-point scale. Tools allowed: N/A Re-exam: Conducted as the ordinary exam

The learning objectives of the courses (knowledge, skills and competencies) and further information on the form and conditions of the assessment can be found in Appendix 1.

9 Workshops

The program covers two workshops, related to PRO1 and PRO2, respectively:

WS1:
The purpose of the workshop is to improve the student's skills in elementary hands-on programming and basic UML modelling, according to the PRO1 curriculum.
To pass the workshop, the student must show an attendance of at least 75% of the lessons during the semester.
Credit is awarded based on a test given in PRO1.

WS2:
The purpose of the workshop is to improve the student's skills related to the PRO2 and SWE1 curriculum.
To pass the workshop, the student must show an attendance of at least 75% of the lessons during the semester.
Credit is awarded based on the grade 10 or 12 in SDJ1 and to students with credit for PRO1.

As a transitional arrangement for students enrolled in 2023, an additional workshop is included in autumn semester 2024.
WS3:
The purpose of the workshop is to gain a basic understanding of computer networks and Internet protocols.
To pass the workshop, the student must show an attendance of at least 75% of the lessons during the semester.
Credit is awarded to students with credit for NES1.

10 Bachelor Project

IT-BPR1 and IT-BPR2

The programme concludes with a bachelor project (BPR2), which accounts for 15 of the programme's total 210 ECTS and concludes with an oral exam. The bachelor project commences in the 6th semester (BPR1) with a choice of subject and preparation of a project description.

The bachelor project must demonstrate independent critical reflection within the chosen topic and must document the student's ability to apply engineering theories and methods. The bachelor project must also reflect the student's ability to express themselves in an academic and structured manner within their subject.

The bachelor project is prepared in groups of 2-3 people. However, the head of programme may grant exemptions from this rule in exceptional circumstances.

The bachelor project includes an independent experimental, empirical and/or theoretical treatment of a practical problem in connection with the central topics of the software technology engineering programme.

The project must be documented in the form of a report containing a project basis, solution description, calculations, drawings, etc. If the report is a group assignment, it must be clear who wrote which sections of the report.

The students are examined in the project by oral exam/group exam with individual assessment in accordance with the programme's overall goals as described in Section 1 of the Curriculum. The basis for exam is the bachelor project. It is a prerequisite for participation in the exam that the bachelor project is submitted within the stipulated deadline and meets the described criteria for the project.

Exam can take place at the earliest when all the other exams of the programme, including internships, have been passed.

11 Title and issue of degree

Graduates who have completed the programme of study according to this curriculum + joint regulations, are entitled to use the Title Bachelor of Engineering in Software Technology.

It is also possible to obtain the following special designations: Interactive Media, Internet of Things or Data Engineering

For completed programmes, VIA University College issues a diploma stating the title, programme and, if applicable, special designation. Furthermore, information is provided on the scope of the sub-elements in ECTS, the result of the assessments achieved as well as the topics for the interdisciplinary project and the graduation project. Similarly, the graduate's basis of admission to the programme is also stated.

In the event of the programme being interrupted, a certificate of passed units of study is issued.

12 Appendix 1: Study plan for intake summer 2023

First and second semester is specified in this appendix.
From third semester the study plan will be as described in this curriculum.

Semester *Theme*	Course 10 ECTS	Course 5 ECTS	Course 5 ECTS	Project 10 ECTS
2 semester ***Client/Server Systems***	**SDJ2** Software Development with UML and Java 2	**SWE1** Software Engineering	**DBS1** Database Systems	**SEP2** Semester Project
1 semester ***Single User Systems***	**SDJ1** Software Development with UML and Java 1	**WEB1** Web Development 1	**DMA1** Discrete Mathematics and Algorithms	**SEP1** Semester Project

See also section 9, Workshops

1st semester: Single User Systems

The goal of the 1st semester is to give the student knowledge and practical skills in object-oriented programming and systems development. The student will also learn about responsive web design, discrete mathematics, and algorithms. The semester is organised with a number of introductory courses and a semester project in which the students will design, implement, and document a single-user software system.

The scope of the semester is 30 ECTS.

The purpose of the courses, ECTS and assessment:

Software Development with UML and Java 1 (SDJ1) – 10 ECTS	**Exam prerequisites**	**Assessment** (all re-exams may be oral exam)
The main purpose of the course is to provide students with the qualifications needed to understand the core object-oriented concepts and to implement smaller programs in Java from UML class diagrams.	1. Attendance (≥ 75%) 2. Participation in oral mid-term test If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual oral exam where the student will pick an unfamiliar programming exercise at random. The student must explain the UML involved and demonstrate how to perform the programming task using a laptop. The time allotted for the exam is 30 minutes including assessment. External assessment. Tools allowed: All. Re-exam: Same as the ordinary exam. Two re-exams (second and third attempt) will be scheduled in the beginning of second semester.

Web Development 1 (WEB1) – 5 ECTS	**Exam prerequisites**	**Assessment**
The purpose of this course is to introduce a set of theories and tools in order for students to obtain a proficient level of knowledge and gain a practical set of skills for designing and developing responsive web sites for both desktops and mobile devices using basic web programming. (WEB1 was previously labelled RWD1, Responsive Web Design, running up to and including spring semester 2022.)	3. Attendance (≥ 75%) 4. Course assignments handed in before deadline and approved. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Digital written exam duration of 2 hours (2 parts): Part 1: Multiple choice questions 30 minutes, weighing 25% Part 2: Short answer questions 90 minutes (explaining and writing code), weighing 75% External assessment. Tools allowed: Part 1: without aids Part 2: all aids allowed – including internet connection. Any types of communication between students or between a student and an external party is prohibited and will be considered a violation of the exam rules. Also, the use of AI tools, such as AI tools, ChatGPT or similar AI and Machine Learning tools and chatbots is not allowed. Re-exam: Re-exams may be oral.
Discrete Mathematics and Algorithms (DMA1) – 5 ECTS	**Exam prerequisites**	**Assessment**
The aim of the course is to train students in the mathematical concepts and process of algorithmic thinking, allowing them to construct simpler, more efficient solutions to real-world computational problems, building on the principles of mathematics.	3. Attendance (≥ 75%) 4. Six out of eight mandatory assignments approved. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: The course is evaluated based on a 3-hour written test. The test is completed in the FLOWlock browser in WISEflow. Internal examiner. Tools allowed: The students are allowed to use any notes, books, and/or other written/printed material. Any type of communication between students or between a student and an external party is prohibited and will be considered a violation of the exam rules. Re-exam: Re-exams may be oral.
Semester Project (SEP1) – 10 ECTS	**Exam prerequisites**	**Assessment**
The purpose is to develop and document a single user system. The PBL purpose is to: • Apply professional competencies in a problem-based context. • Solve engineering problems based on the semester courses. • Demonstrate the ability to prioritize between problems and to work in detail on selected problems.	None	Type of exam: Group exam with individual assessment. Exam is based on the project report and process report, which must be submitted before deadline and apply with the formalities criteria stated under the Software Engineering specific Guidelines. Group presentation of 5 minutes times the number of group members followed by a joint exam of 15 minutes times the number of group members including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the exam. Internal assessment. Tools allowed: All.

The focus of the PBL teaching in SEP1 is LEARNING TO LEARN, project methodology and PBL, including a basic introduction to study techniques and team-based project work.		Re-exams: Students who failed a semester-project must make a new project. Students who fail a semester project must attend an information meeting at the end of the summer exam period. At this meeting, students will be notified about the process of the re-exam and students will form groups. Deadline for hand in of the project is mid-August (exact date will be informed at the meeting). There will be no guidance in the period up to hand in. Oral assessment of the project takes place before the start of the autumn semester.

The learning objectives of the courses (knowledge, skills, and competencies) can be found in Appendix 2.

2nd semester: Client/Server Systems

During the 2nd semester, the student will learn about software engineering methods, database development and system development methods, and the student will accumulate the programming skills necessary to build client/server systems. The semester is organised as a number of compulsory courses, along with a semester project in which the students will develop a client/server system.

The scope of the semester is 30 ECTS.

The purpose of the courses, ECTS and assessment:

Software Development with UML and Java 2 (SDJ2) – 10 ECTS	**Exam prerequisites**	**Assessment** (All re-exams may be oral exam)
The purpose is to qualify the student to understand and master the concepts and techniques of object-oriented system development and programming, including Client/Server programming.	Attendance (≥ 75%) If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual oral exam, 20 minutes, without preparation based upon various course assignments, which must be submitted before the given deadline. • The student will draw from a pool of previously known questions. • The student will explain concepts and theories from the course, using the course work as reference. • The student will start with a prepared presentation. External assessment. Tools allowed: N/A Re-exam: Same as the ordinary exam.
Database Systems (DBS1) – 5 ECTS	**Exam prerequisites**	**Assessment**
The course has two purposes. Firstly, students are to learn methods for designing, implementing and operating single-user relational databases. Secondly, students are to learn the main principles, architecture and technologies of a typical relational database management system (RDBMS).	1. Attendance (≥ 75%) 2. Course assignments handed in before deadline. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: The students must import a database and dataset before the exam. The first part of the exam will ask questions about this dataset. This dataset will be made available through WISEflow several days before the exam. Digital written exam (2 parts), 4 hours: Part 1: Multiple choice and written answers in WISEflow Part 2: Design and implementation External assessment Tools allowed: All aids are allowed included access to online material. However, it is not allowed to use AI tools such as ChatGPT and similar AI and Machine Learning driven tools and chatbots.

		Re-exam: Same as the ordinary exam.
Software Engineering (SWE1) – 5 ECTS	**Exam prerequisites**	**Assessment**
The purpose is to qualify the student to apply software engineering concepts used to develop object-oriented software. Structure the software development process by applying SCRUM and Unified Process to conduct Analyse, Design and Test-descriptions to exemplify a final solution from a real-life problem. This involves requirement capturing (Use Cases and non-functional requirements), analysis, domain models, interaction diagrams, design classes, design patterns and test-descriptions etc.	2. Attendance (≥ 75%) 3. Course assignments handed in before deadline. If the exam prerequisites are not met, the student must complete a written assignment in WISEflow to qualify for the re-exam. This assignment will be scheduled after the ordinary exam.	Type of exam: Individual oral exam, 20 minutes. Exam is without preparation and based on a drawn question. The questions will be known before the exam. Internal assessment. Tools allowed: All Re-exam: Conducted as the ordinary exam. **NOTE!** Appendix 2 describes the exam form that was valid in 2022 and in spring 2023.
Semester Project (SEP2) – 10 ECTS	**Exam prerequisites**	**Assessment**
The purpose is to develop and document a client/server system. The purpose of the PBL part of the course is to promote the students' competencies in collaboration, planning and problem analysis.	None	Type of exam: Group exam with individual assessment. Exam is based on the project report and process report, which must be submitted before deadline and apply with the formalities criteria stated under the Software Engineering specific Guidelines. Group presentation approx. 20 minutes followed by a joint exam with a joint discussion and individual question rounds for approx. 20 minutes per student including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the exam. Internal examiner. Tools allowed: All. Re-exams: Students who failed a semester must make a new project. Students who fail a semester project must attend an information meeting at the end of the summer exam period. At this meeting, students will be notified about the process of the re-exam and students will form groups. Deadline for hand in of the project is mid-August (exact date will be informed at the meeting). There will be no guidance in the period up to hand in. Oral assessment of the project takes place before the start of the autumn semester.

The learning objectives of the courses (knowledge, skills and competencies) can be found in Appendix 1.

13 Appendix 2: Courses Software Technology Engineering Programme

Code	Title	ECTS-points	Knowledge	Skills	Competencies
IT-PRO1	Programming 1	10	The student should be able to: - Identify the Java lexical structures: keywords, separators, operators, identifiers, literals and comments. - Explain details of UML class diagrams. - Identify selection and loop structures in UML activity diagrams.	The student should be able to: - Construct Java programs with proper choice of selection and loop structures. - Create and use objects in Java. - Implement classes in Java using the object oriented concepts: encapsulation, inheritance and polymorphism. - Implement one-to-one relations and differentiate between association, aggregation and composition. - Implement one-to-many relations using array structures and a simple collection class. - Implement exception handling for different types of exceptions. - Implement persistence in text and binary files. - Construct simple event-based GUI applications. - Construct Java source code documentations. - Interpret UML class diagrams, and construct corresponding Java code.	The student should be able to: - Exemplify and discuss basic object-oriented concepts, including encapsulation, relationships, inheritance and polymorphism - Implement small scale systems from UML class diagrams.
IT-WEB1	Web Development	5	Having completed this course, students will have the knowledge to: - Describe the different file formats used in web development and their purpose . - Reproduce webpage layouts using HTML5 and CSS3 when presented with images/screenshots of other websites . - Select appropriate attributes for HTML5 elements . - Explain the difference between responsive and non-responsive websites. - Test HTML5 files for errors using the W3C markup validator . - Account for the difference between the JavaScript and Java programming languages .	Having completed this course, students will have the skills to: - Create web sites using Hyper Text Markup Language (HTML5) . - Use simple and advanced CSS3 selectors and properties to style webpages . - Apply the Bootstrap grid framework to create responsive websites . - Utilize the Bootstrap classes to apply styling to responsive websites . - Implement JavaScript functions to add functionality to websites . - Use XMLHttp Request to read content from an external source and integrate this content into a website . - Select HTML elements and apply jQuery	Having completed this course, students will be able to: - Design and implement platform independent web applications .

Code	Title	ECTS-points	Knowledge	Skills	Competencies
				animations to the selected elements to make websites interactive .	
IT-MSE1	Mathematics for Software Engineering	5	After completing the course, students will be able to: - Explain the principles of basic arithmetic and number systems, essential for all computational tasks. - Describe Boolean algebra's applications in logic circuits and programming conditionals. - Discuss combinatorial problems and basic problems in probability theory - Summarise the concept of linear equations, solution sets, and linear dependence in the context of data structures and algorithm optimization. - Explain matrix algebra, its significance in complex computations, and applications in software technology - Examine the application of eigenvalues and eigenvectors in software technology - Outline the fundamentals of asymptotic analysis and Big O notation for evaluating algorithm efficiency. - Account for the concept of functions and loops in programming, emphasizing their importance in creating efficient and reusable code.	Upon course completion, students will gain skills in: - Performing arithmetic operations and utilizing different number systems for varied computational contexts. - Applying Boolean algebra in developing and optimizing logical expressions and algorithms. - Solving combinatorial problems and employing probability theory to analyze and predict outcomes in software projects. - Utilizing conditional probability and Bayes' Theorem - Solving linear equations, understanding solution sets and linear dependence to enhance data analysis and algorithm development. - Manipulating matrices and applying matrix algebra in transformations and complex calculations. - Determining eigenvalues and eigenvectors - Applying asymptotic analysis and Big O notation to assess and improve algorithm performance. - Creating functions and implementing loops in programming to improve code efficiency, readability, and maintainability.	After the course, students will be competent in: - Formulating and solving linear and combinatorial problems analytically and algorithmically, preparing them for complex challenges in software engineering. - Designing and optimizing algorithms using Boolean algebra, matrix operations, and asymptotic analysis, crucial for high-performance software development. - Developing the basis for predictive models and decision-making systems by applying concepts of probability, conditional probability, and Bayes' Theorem. - Applying the acquired mathematical skills and programming techniques in more advanced software engineering courses and real-world projects, enabling them to tackle complex problems with confidence.
IT-DMA1 (in A23)	Discrete Mathematics and Algorithms	5	Upon completion of this course, students will be able to: - Describe fundamental concepts in number theory and modular arithmetic - Outline the basic principles of different sorting algorithms - Summarize key aspects of various data structures	Upon completion of this course, students will be able to: - Give precise arguments for the correctness or incorrectness of an algorithm - Use key concepts of discrete mathematics for solving programming problems resourcefully - Analyse and compare the time and space usage of algorithms and data structures	Upon completion of this course, students will be able to: - Adapt known algorithms and data structures to special cases of known problems or new problems - Design and implement small programs, using algorithms and data structures taught in the course . - Evaluate the performance of Java code with the objective of designing and implementing algorithms that optimise the code
IT-SDJ1 (in A23)	Software Development	10	The student should be able to: - Identify the Java lexical structures: keywords, separators, operators, identifiers, literals and	The student should be able to: Construct Java programs with proper choice of selection and loop structures .Create and use objects in Java	The student should be able to: - Exemplify and discuss basic object-oriented concepts, including encapsulation,

Code	Title	ECTS-points	Knowledge	Skills	Competencies
	with UML and Java 1		comments. - Explain details of UML class diagrams. - Identify selection and loop structures in UML activity diagrams.	.Implement classes in Java using the object oriented concepts: encapsulation, inheritance and polymorphism .Implement one-to-one relations and differentiate between association, aggregation and composition .Implement one-to-many relations using array structures and a simple collection class .Implement exception handling for different types of exceptions .Implement persistence in text and binary files .Construct simple event-based GUI applications .Construct Java source code documentations . Interpret UML class diagrams, and construct corresponding Java code .	relationships, inheritance and polymorphism - Implement small scale systems from UML class diagrams .
IT-SEP1 (in A23)	Semester Project: Single User System	10	The student will use the knowledge acquired in SDJ1, WEB1 and DMA1.	Professional Skills Learning objectives: - Explain the Waterfall method as a software development process - Derive requirements- Apply use case modelling and draw activity diagrams - Draw a domain model - Construct UML class diagram(s) - Draw a sequence diagram of one essential method - Implement a software system using object-oriented programming - Integrate Java-generated files into a webpage using JavaScript - Perform testing in relation to the derived requirements - Describe your system in a user guide - Reflect on the waterfall method as a tool to control a software development PBL Skills Learning Objectives: - Establish and contribute to a collaboration with the group and the supervisor - Apply covered theories on group dynamics, teamwork and conflict resolution - Formulate and enforce a group contract with the group. - Identify relevant problems, formulate a problem statement, and explain proposed	Professional Competency Learning Objectives: - Demonstrate the connection between the different disciplines in software development - Evaluate the performance of selected parts of the system in terms of time and space complexity using the Big O notation PBL Competency Learning Objectives: - Apply and reflect on covered theories on learning and motivation. - Describe and reflect on the group's cooperation.

Code	Title	ECTS-points	Knowledge	Skills	Competencies
				solutions. - Apply knowledge about references and source management. - Apply academic and technical writing style, report structure and rules for plagiarism. - Communicate the results of the project work in writing, graphically and orally to different target groups.	
IT-SEP1	Semester Project: Single User System	10	The student will use knowledge acquired in PRO1, MSE1 and WEB1.	The student should be able to: - Make a project description with documentation of problem domain - Derive requirements - Apply use case modelling and draw activity diagrams - Construct a domain model - Construct UML class diagram(s) - Construct a sequence diagram of one essential method - Implement a software system using object-oriented programming - Integrate Java-generated files into a webpage using JavaScript - Perform testing in relation to use cases - Describe your system in a user guide - Apply relevant theories on group dynamics, team cooperation and conflict resolution and formulate and enforce a group contract - Communicate the results of the project work using academic and technical writing, apply the correct report structure and rules on plagiarism - Communicate successfully in writing, graphically and orally to different target groups - Evaluate the performance of selected parts of the system in terms of time complexity using the Big O notation.	The student should be able to: - Demonstrate the connection between the different disciplines in software development - Describe and reflect on the group's cooperation - Apply and reflect on covered learning theories and motivation theories - Reflect on the waterfall method as a tool to control a software development process.
IT-PRO2	Programming 2	10	The student should be able to understand: - System architecture - Various methods for testing - Concurrent programming - Design patterns - Client/server structure - Layered abstractions in protocol stacks	The student should be able to understand: - System architecture - Various methods for testing - Concurrent programming - Design patterns - Client/server structure - Layered abstractions in protocol stacks	The student should be able to: - Implement programs in Java using design patterns, and evaluate which to use - Test software using relevant testing techniques - Develop flexible java code using interfaces - Implement thread-safe classes and multi-

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			- The Internet's naming system - Addressing in the Internet	- The Internet's naming system - Addressing in the Internet	threaded programs - Implement client-server systems - Create application layer protocols for distributed systems
IT-SWE1	Software Engineering	5	The student should be able to account for: - Abstraction - UML (selected diagrams) - S. O. L. I. D principles - Unified Process - Scrum - Design principles - Architectural design - Requirement capturing - Analysis vs. Design models - The difference between software development and coding - Test descriptions - How to conduct a test following a test description	The student should achieve the skills to: - Analyse a problem and document the analysis- and design-process with text and UML - Apply use of Scrum - Apply use of Unified Process - Use UML to document requirements, analysis, and design artefacts - Use agile software development with Unified Process in combination with Scrum - Create a Domain model from a problem description, requirement specification and understand the elements in the resulting Domain model - Create a design model and understand the elements within it - Apply the S .O .L .I .D principles on a design model - Design for test - Create test descriptions - Create architectural design models	The student should be able to: - Analyse a problem– what is the problem to be solve? - Derive a requirement specification with Use Cases and non-functional requirements - Plan tests by Test specifications - Analyse and design a project to be implemented in teams with many participants and stakeholders - Work in a Scrum team
IT-DBS1	Database Systems	5	Having completed this course, students will be able to: - account for the relationship between relational algebra and SQL - define the relational model - distinguish the 3 normal forms - classify keys in relational databases - explain indexes	Having completed this course, students will be able to - use relational modelling to model business cases - use UML to document ER-Models - use Data Definition Language (DDL) to create databases - use Data Modelling Language (DML) to manipulate data in a database - use the mapping method to convert ER-Models to Relational Models - use normalisation to normalise a database schema to 3rd normal form - create SQL statements to create, replace, update and delete data in a database - use keys in relational databases - use joins to combine data	Having completed this course, students will be able to: - Design and implement a database schema on the 3rd normal form - Use a database in application development

Code	Title	ECTS-points	Knowledge	Skills	Competencies
				- use transactions to prevent data corruption - create triggers - create views	
IT-SDJ2 (in S24)	Software Development with UML and Java 2	10	The student should be able to understand: - System architecture - Various methods for testing - Concurrent programming - Design patterns - Client/server structure	The student should achieve the skills: - Implement design patterns in Java - Test software using different testing techniques, including (but not limited to) Junit testing, System testing, etc . - Implement thread-safe classes and multi-threaded programs - Make programs communicate using client-server technologies	The student should be able to: - Implement programs in Java using design patterns, and evaluate which to use - Test software using relevant testing techniques - Develop flexible java code using interfaces - Implement thread-safe classes and multi-threaded programs - Implement client-server systems
IT-SEP2	Semester Project: Client/Server System	10	Apply all knowledge from all 2. Semester courses	Professional skill learning objectives: - Document the analysis and design of a software system using UML. - Devise requirements, use case model, activity diagrams, system sequence diagrams, domain model. - Design and implement (using SQL) a normalized relational database for data-persistence. - Apply unit and use case testing based on requirements and code. - Implement a client/server software system in Java according to design. - Apply Scrum in combination with Unified Process (UP) and relevant tools in the project work. PBL skill learning objectives: - Be able to enforce and develop the group contract - Can work based on the project group's own problem statement. - Apply covered theories on personal profiles and cross-cultural aspects in the group. - Apply knowledge about references and source management. - Apply academic and technical writing style, report structure and rules for plagiarism. - Communicate the results of the project work and learning process of the project group in a	Professional skill learning objectives: - Document the analysis and design of a software system using UML. - Devise requirements, use case model, activity diagrams, system sequence diagrams, domain model. - Design and implement (using SQL) a normalized relational database for data-persistence. - Apply unit and use case testing based on requirements and code. - Implement a client/server software system in Java according to design. - Apply Scrum in combination with Unified Process (UP) and relevant tools in the project work. PBL skill learning objectives: - Be able to enforce and develop the group contract - Can work based on the project group's own problem statement. - Apply covered theories on personal profiles and cross-cultural aspects in the group. - Apply knowledge about references and source management. - Apply academic and technical writing style, report structure and rules for plagiarism. - Communicate the results of the project work and learning process of the project group in a

Code	Title	ECTS-points	Knowledge	Skills	Competencies
				well-structured manner using technical terms in writing, graphically and orally.	well-structured manner using technical terms in writing, graphically and orally.
IT-PRO3	Programming 3	5	The students will be able to - explain various distributed system architectures - explain various distributed communication methods - explain the use of contracts in service calls - identify common Internet threats - describe common access control systems e.g., packet filter, proxy, etc. - describe privacy, integrity, and authentication methods.	The students will be able to - use various distributed communication methods for direct and indirect communication - argue the choice of middleware for a given distributed system - compare and contrast different encryption technologies - discuss how confidentiality, integrity and availability can be accomplished using security technology.	The students will be able to - design the architecture of a distributed system using known architectural patterns - design and implement a distributed system on different platforms using various middleware - identify security threats and propose mechanisms to mitigate these threats.
IT-CAO1	Computer Architecture and Organisation	5	Having completed this course, the student is able to: - Describe and apply numbering representations, including two's complement to represent negative numbers in the binary numbering representation - Identify the functionality of basic logic gates and be able to combine them into half- and full-adders, flip/flops, etc. - Describe Boolean algebra and its relation to digital circuits - Describe the architecture of simple CPUs and how they function, explain the build and working behaviour of basic building blocks of CPUs (registers, ALUs, etc.)	Having completed this course, the student should be able to: - Create functioning assembler programs for microcontrollers - Analyse ASM programs (AVR MCU) and calculate execution time - Execute and debug assembler programs - Analyse and describe simple logical circuits (Boolean expressions) - Apply Boolean algebra to reduce digital circuits.	Having completed this course, students should be able to: - Describe the functionality of the components of basic computer architectures - Apply mathematical theory to understand low-level computer architecture and programming - Create simple logic circuits used in CPUs - Create applications using assembler programming - Integrate simple I/O devices in embedded applications.
IT-DNP1	.NET Programming	5	The student will be able to: - Describe the fundamentals of .NET development and the common type system - Identify and describe .NET technologies relevant to web application development	The students will be able to: - Write and debug C# code - Implement RESTful Web Services in relation to a distributed system - Consume RESTful Web Services - Utilize asynchronous programming - Create and interact with a relational database using an Object - Relational Mapping library - Define and implement basic authentication and authorization - Navigate and use the managed .NET API - Create and consume class libraries - Compare object-relational mapping to	The student will be able to: - Implement a robust, error-safe system - Implement console applications, web applications and web services as part of a distributed system with - Server-side and client-side C#-programming - Data persistence using object-relational mapping - User management, including authentication and authorization - Analyze and evaluate the relevance of .NET technologies when designing software applications - Apply best practices when developing .NET

Code	Title	ECTS-points	Knowledge	Skills	Competencies
				traditional data access techniques - Implement a Web App within ASP.NET	apps
IT-ADS1	Algorithms and Data Structures	5	Upon completion of the course, the student should have knowledge about - Time complexity of algorithms - Various linear and non-linear data structures - Abstract datatypes - Various types of algorithms and their applicability	Upon completion of the course, the student should be able to - analyze algorithms using big-Oh notation - design and implement algorithms and data structures in an object-oriented programming language	Upon completion of this course, the goal is that the students have acquired the competences to: - assess which type of algorithm is best suited for a given problem - assess which data structures are best suited in a given context
IT-SEP3	Semester Project: Heterogeneous System	10	Can refer to involved theories in order to increase efficiency for the group as a whole but also for the individual student.	In addition to the skills acquired in IT-SEP2, the student will be able to: - Implement heterogeneous systems using multiple network protocols - Analyse the security risks of a distributed system - Use a version control system to manage versions PBL Skills Learning Objectives - Search for, locate, and apply relevant knowledge. - Apply academic and technical writing style, report structure, and rules of plagiarism. - Communicate the results of project work and the learning process of the project group in a well-structured manner using technical terms in writing, graphically and orally.	In addition to the competences acquired in IT-SEP2, the student will be able to: - Analyse, design, implement and test a distributed system using UML, Java and C# - Construct a distributed system with the proper selection of architectural patterns - Argue for the choice of various technical solutions for implementing distributed systems PBL Competency Learning Objectives - Structure and adapt group collaboration to the preferences and competencies of the members. - Receive and reflect on guidance and facilitation of group collaboration. - Independently plan, structure, and optimize own learning process based on previous experiences. - Argue for the choice of sources, methods, and solutions based on a critical assessment. - Incorporate a holistic and sustainable approach to the project with an eye for connections to the surrounding world.
IT-MAL1	Introduction to Machine Learning and AI	5	After having successfully completed the course, the student will have gained knowledge about algorithms, methods, techniques, tools, and applications within the following machine learning methods: • Different data preparation and preprocessing methods • Different types of classification algorithms, e.g., Naïve Bayes, k-Nearest Neighbor,	After successfully completing the course, the student will have developed the following skills: • Ability to prepare and preprocess data for various machine learning applications effectively. • Proficiency in implementing and tuning classification algorithms and selecting the appropriate classifier for a given dataset.	Upon completion of the course, students are expected to have acquired the competences to: • Make informed decisions regarding the selection and application of machine learning techniques tailored to specific problem domains. • Fine-tune and parametrize machine learning algorithms to optimize their performance on specific datasets.

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			Decision Trees, Logistic Regression, Support Vector Machines, Neural Networks • Different types of regression algorithms, e.g., simple linear regression, multiple linear regression, Ridge regression, Lasso regression • Different types of dimensionality reduction algorithms, e.g., Principal component analyses, singular value decomposition, factor analysis • Different types of clustering algorithms, e.g., k-Means clustering, Agglomerative clustering, DBSCAN • Different metrics for assessing the strength and quality of their machine learning algorithms	• Capability to apply regression techniques to predict continuous variables and evaluate the predictive ability of regression models. • Using dimensionality reduction algorithms to interpret and simplify complex datasets. • implementing clustering algorithms to categorize unlabelled datasets and determining the optimal number of clusters. • Ability to use various machine learning tools and libraries • Critical evaluation of model performance using various metrics and validation techniques.	• Conceptualize, design, and develop machine learning solutions for real-world problems. • Articulate, communicate, and deliberate machine learning solutions, their implications, and associated decisions with both domain experts and non-technical persons.
IT-ESW1	Embedded Software	5	The student should be able to account for: - Analysis of an embedded problem and documentation of the analyse- and design-process in UML - Dynamic memory management in C - Unit test of C-programs	The student should be able to exemplify: - Analyse and design of an embedded software solution - Implement functioning programs in C - Implement programs in C using different API's and libraries for hardware drivers etc.	The student should be able to: - Design and construct software for an embedded systems using C-programming
IT-WEB2	WEB Development 2	5	After successfully completing the course, the student will have gained knowledge to: - Explain scope and closures in JavaScript - Compare dynamically and statically typed languages - Describe the JavaScript object model - Explain how 'this' works in JavaScript - Outline how prototypes, constructors & the class keyword are used in creating JavaScript objects - Describe how modules work in JavaScript - Compare the use of object-oriented and functional programming paradigms in JavaScript - Explain how concurrency works in JavaScript - Describe how unidirectional data flows are used in front-end applications - Explain how state is managed in a front-end application	After successfully completing the course, the student will have acquired the skills to: - Manipulate web pages using JavaScript - Use various JavaScript expressions and operators such as destructuring assignment, spread syntax, rest parameters, short circuit operators and optional chaining - Utilize factory functions to create objects in JavaScript - Make use of concatenative and prototypal inheritance in JavaScript - Apply higher-order functions to abstract over actions - Use callbacks, promises and async/await for asynchronous programming - Organize and clarify code with object-oriented and functional programming techniques - Consume web services using fetch & XMLHttpRequest	After successfully completing the course, the student will have acquired competencies in analyzing, designing and constructing web applications using JavaScript and modern front-end frameworks.

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			- Compare client-side-, server-side- and static rendering patterns - Summarize the basics of TypeScript	- Enhance the development process of web applications with built tools like package managers, module bundlers, preprocessors and task runners - Built single page web applications using React - Work with client side routing in a web application - Test the functionality of web applications using unit-, integration- and end-to-end tests - Apply patterns and best practices to measure and improve web performance - Implement a Node web API using Express - Work with authentication using JSON Web Tokens	
IT-DOC1	DevOps & Cloud	5	The student should: - Understand the full software development life cycle. - Understand key DevOps concepts. - Know about the general categories of tools that can be used for automating workflows and integrating the processes of software development and IT operations.	The student should be able to: - Explain the role & purpose of the specific tools introduced in the course. - Explain general DevOps concepts and tools and how they help to develop software faster and more reliably. - Assess various architectural approaches for making software easier to develop, test, maintain and deploy. - Explain the general workings of the specific tools introduced in the course and how they work together to make software development more efficient. - Use a version control tool and integrate the use thereof in the full software development life cycle. - Use a tool/technology for encapsulating an executable together with all its dependencies in an easy to deploy container that is decoupled from the specifics of the operating system.	The student should be able to: - Identify and maintain key development and deployment configuration data as versioned code and automate the recreation of a system setup from such versioned configuration data. - Demonstrate ability to apply and use DevOps tools and methods in their semester project.
IT-SEP4	Semester Project: Internet of Things	10	The student will apply the knowledge acquired in WEB2, ESW1 and DOC1. Furthermore, the student will gain knowledge about IoT-hardware and sensor/actuator API used in the project.	After successfully completing the course, the student will have acquired the skills to: - Write, test and deploy code for custom IoT-hardware - Define interfacing contracts for larger software systems	After successfully completing the course, the student will have acquired competencies to: - Communicate and coordinate in larger software teams - Analyze and design complete solutions comprising of both hardware and software

Code	Title	ECTS-points	Knowledge	Skills	Competencies
				- Utilize a cloud provider to host parts of a larger software project - Automate the software delivery pipeline through various DevOps approaches and tools - Setup and maintain a build server for a larger software project - Setup and maintain automated regression testing - Design and implement applications in C - Design and implement web applications in JavaScript - Devise requirements, use case model, activity diagrams, system sequence diagrams, domain model - Document the analysis and design of a software system using UML - Carry out a software project in close cooperation with the group - Apply Scrum in combination with Unified Process and relevant tools in the project work PBL Skills Learning Objectives Apply academic and technical writing style, report structure and plagiarism rules.	- Decide on appropriate quality assuring methods for a given software development project - Implement full-scale IoT-solution - Conduct projects in multidisciplinary teams - Work analytically, methodically and structured with the semester project in the project group - Carry out a software project with a clear connection between requirements, analysis, design, implementation, test, and documentation PBL Competency Learning Objectives - Plan, structure and execute effective interdisciplinary collaboration. - Reflect on knowledge sharing in the project group and with other groups. - Communicate and argue for the results of the project work and the project group's learning process in a structured way using academic concepts, both in writing, graphically, and orally. - Argue for the choice of sources, methods and solutions based on a critical assessment. - Explain ethical considerations in the project work.
IT-INP1	Engineering Internship (IT-)	30	The student must: - gain knowledge of theory, methodology and practice within a profession or one or more fields of study - be able to understand and reflect on theories, methodology and practice - be aware of non-technical – societal, health and safety, environmental, economic and industrial – implications of engineering practice.	The student must: - be able to apply the methodologies and tools of one or more fields of study and to apply skills related to work within the field/fields of study or profession - be able to assess theoretical and practical problems and to substantiate and select relevant solutions - be able to communicate professional issues.	The student must: - be able to handle complex and development oriented situations in study or work contexts - be able to independently participate in professional and interdisciplinary collaboration with a professional approach - be able to identify own learning needs and to organise own learning in different learning environments - promote an engineering-oriented approach during the remaining semesters on the Bachelor programme - develop personal skills required for the professional career as engineer - form the basis for developing personal/professional network

Code	Title	ECTS-points	Knowledge	Skills	Competencies
ENG-IDE1	Innovation and Entrepreneurship project	10	After having successfully completed the course, the students will have gained: - An understanding of innovation and entrepreneurship and its uses within the field of engineering and business. - Knowledge about three different innovation processes Design Thinking, Effectuation and Lean Startup - Knowledge about how to create a systematic and measurable progress in innovation and entrepreneurship tasks	After having successfully completed the course, the students will be able to: - Engage in innovative and entrepreneurial processes in a cross-discipline setting - Conceive, plan, and execute innovative ideas - Work methodically with innovation and entrepreneurship - Collect and apply relevant data/information about technologies, markets, and end users - Apply method to gain insights about the solutions impact on the current market. - Convey and argue for the results of a cross-disciplinary project group and the project group's learning process using correct professional terminology and optimal tools both in writing, graphically and orally.	After having successfully completed the course, the students will have gained competences in: - Introducing innovative ideas into project work - Contributing own professional skills in multidisciplinary teams with the objective of solving problems by using innovative and entrepreneurial processes and models - Clarifying multidisciplinary group competencies - Analyzing group dynamics and adapting working methods and collaboration methods to new group constellations to achieve effective collaboration in cross-disciplinary project teams - Independently structuring and planning own learning process in an interdisciplinary learning environment Able to independently argue for the application and implementation of valid knowledge
IT-SPRAU	Semester Project (short course)	10	After having completed this course, the students must master the knowledge about: • Searching and scoping relevant project information • Project and team work planning • Communication and documentation skills • Testing	After having completed this course, the student must be able to: • Identify and justify problems and their context • Select and argue for choice of method and reflect critical on methods • Find and assess relevant literature within the problem domain • Present the result for an audience of engineers	After having completed this course, the students must be able to: • Describe and delimit a large Software Engineering Project • Select and use relevant theories and methods to solve the problem • Plan and structure the project within the project time frame • Initiate the preliminary steps in a system development process, leading to a clearly defined requirements capture, use cases as well as object and behaviour analysis. • Work successfully in a project group with the objective of solving a well-defined engineering problem.
IT-BPR1	Bachelor Project Preparation	5	After having completed the course, the student should be able to - Explain the concept of plagiarism and how to avoid it. - Identify a problem and a problem domain in software engineering. - Evaluate teamwork and team dynamics.	After having completed the course, the student should be able to - Perform information search and retrieval. - Describe a proposed problem to solve, as well as its context. - Plan and delimit a software development project.	After having completed the course, the student should be able to - Plan a larger Software Engineering project and describe it in a Project Description. - Apply preliminary steps in a system development process.

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			- Select relevant methods for developing a project. - Grasp essential concepts in the philosophy of science, including knowledge, paradigms, pseudo-science, and non-science. - Gain a basic understanding of ethics and ethical thinking within the scope of science.	- Assess the relationship between scientific knowledge and practical experience in technology creation. - Collect data from users, considering ethical implications. - Effectively interact with others, enhancing their communication, empathy, and teamwork abilities. - Use self-reflection to collaborate effectively in team environments.	- Develop clear and concise requirements using a selected standard. - Demonstrate the ability to work coherently in a group. - Reflect upon their own role as knowledge creators in an information society.
IT-BPR2	Bachelor Project 2	15	After having completed this course, the students must master the knowledge about: - Searching and scoping relevant project information - Project and teamwork planning - Communication and documentation skills - Testing	After having completed this course, the student must master to: - Identify and justify problems and their context - Select and argue for choice of method and reflect critical and said methods - Find and assess relevant literature within the problem domain - Present the result for an audience of engineers	After having completed this course, the students must be able to: - Describe and delimit a large Software Engineering Project - Select and use relevant theories and methods to solve the problem - Plan and structure the project within the BPR2 time frame - Initiate the preliminary steps in a system development process, leading to a clearly defined requirements capture, use cases as well as object and behavior analysis. - Work successfully in a project group with the objective of solving a well-defined engineering problem.
IT-ALI1	Applied Linear Algebra	5	After successfully completing the course, the student will have gained knowledge about: - What a vector space is, and How a linear representation of such spaces can be analysed using matrix operations - Application of linear algebra in engineering	After successfully completing the course, the student will be able to: - Apply techniques and results from linear algebra to solve problems in linear systems, matrices, vector space, orthogonality, eigen vectors, and eigenvalue - Apply theory to analyse basic theoretic tasks within the below mentioned topics - Express mathematically correct arguments - Use mathematical terminology and symbol language	After successfully completing the course, the student will have acquired competences in: - Applying linear algebra to the study of various phenomena in engineering science - Using matrices to solve concrete problems - Using vector operations to solve concrete problems - Applying methods and results from linear algebra in the solution of engineering problems
IT-BUI1	Business Intelligence	5	Students will obtain knowledge about understanding, reading, and presenting data from a dimensional model (such as a star schema or data cube) and other data models. - Knowledge about building data products for operational vs real-time systems	- Data migration using data integration tools - Create Data pipelines to cleanse data and move it into a data warehouse - Create KPIs and measures - Create data analyses, presentations and dashboards with Business Intellligence tools	- Evaluate pros/cons of different BI products, architectures and approaches

Code	Title	ECTS-points	Knowledge	Skills	Competencies
				- Create data structures for analysis purposes with selected tools - Create, deploy and manage reports	
IT-CAL1	Calculus	5	After having successfully completed the course, the student will have gained knowledge about the theory, techniques and tools of calculus, in particular knowledge about: - Functions - Limits and continuity - Derivatives - Integrals - Infinite series and sequences - Partial derivatives - Multiple integrals - Differential equations	Upon completion of this course, students will be able to: - Define and interpret functions, including calculating limits of functions and the concept of continuity - Calculate and interpret ordinary and partial derivates of real functions. - Calculate and interpret definite and indefinite integrals of real functions. - Perform calculations pertaining to infinite series and sequences. - Solve differential equations.	Upon completion of this course, the goal is that the students have acquired the competences to: - Make informed choices about the use of differential and integral calculus. - Apply the tools of calculus to real-world problems. - Communicate and discuss the theory, tools and techniques of calculus.
IT-CMC1	Compiler Construction	5	The students will be able to - describe the main purposes of a compiler - explain the differences between syntax and semantics of a programming language - explain context free grammars - list examples of common programming language features	The students will be able to - construct a context free grammar for a programming language - define the semantics of a programming language in an informal way - design the runtime organisation for a programming language	The students will be able to - design a small, simple programming language - design and implement a compiler for a small, simple programming language using various design patterns and an object oriented language for implementation
IT-DAI1	Data Analytics Infrastructure	5	Having completed this course, students should be able to describe basic techniques within the field, and argue the choice and applicability of these for different use scenarios. This includes: - Application of analytical data processing, and differences to transactional processing - Types of analytical data processing, such as reporting and visualization - Sources of data for analytical processing - Server and locally hosted platforms for data storage and analytical processing - Modelling techniques for designing data models for integration of multi-source data, including structured, semi-structured and unstructured data, and for modelling time-variant data/history - Design of systems for data acquisition,	Having completed this course, students should be able to: - Design and implement data models for integrating multi-source data, including dimensional data modelling, for structured and semi structured data - Design and implement data models for time-variant data - Design, implement and test systems for data acquisition, validation, integration and delivery from multiple sources and platforms - Design, implement and test basic descriptive statistical analysis on integrated data - Design, implement and test basic visualizations and graphs of data and analysis results. - Give relevant peer feedback on handins and exercises throughout the semester	Having completed this course, students should be able to - Discuss and argue pros, cons and trade-offs of choices - Use basic statistics and visualization to find and explain patterns of information in data - Evaluate and act upon peer feedback

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			validating and cleansing data, integration and publishing of data.		
IT-DCA1	Domain Centric Architecture	5	The student should be able to understand: - Common architectural styles, among other: n-layered, vertical slices, hexagonal, clean, etc. - Different methods for testing - Several architectural patterns - Selected strategic and tactical patterns from Domain Driven Design - Basics of Web API - Basics of Object-Relational Mapping - How to develop a system through automated test	The student should achieve the skills: - Apply architectural patterns in practice to build robust systems - Apply architectural concepts - Split a system by various concerns - Test software using automated tests (unit-/integration-tests). - Design and develop a rich, behaviour-driven domain model, which reflects business-logic - Apply tactical patterns from Domain Driven Design - Effectively use an object-relational mapper for both database-generation and code-scaffolding - Implement an RPC-oriented Web API	The student should be able to: - Implement programs while considering architectural styles, ensuring low coupling, high cohesion, and clear separation of concerns - Discuss different architectural styles, highlighting their strengths, weaknesses - Critically reason about and practically implement key architectural patterns, understanding their implications and benefits - Evaluate and reason about different Web API patterns - Reason about dependencies between system-components - Thoroughly analyze complex domains to develop rich domain models that accurately represent business logic and requirements
IT-DIM1	Digital Multi Media	5	Having completed this course, students should have profound knowledge of: - Computer Graphics - Design Principles for multimedia - Video, Animation and Sound - XML and Multimedia		
IT-DSP1	Digital Signal Processing	5	After successfully completing the course, the student will have gained knowledge about: - The nature and recording of different types of digital signals - Cleaning up digital signals - Extracting useful values from digital signals - MATLAB as a tool for development of signal processing algorithms	After successfully completing the course, the student will be able to: - Record digital signals - Applying different filters (high-pass, low-pass, band-pass, notch) to remove unwanted components of digital signals - Use the Fast Fourier Transform to analyze the frequency content of a signal	After successfully completing the course, the student will have acquired competences in: - Explain sampling processes and how to determine the correct sampling frequency - Describe signal processing applications - Applying digital signal processing methods to analyze and interpret engineering problems - Develop signal processing algorithms
IT-EOS1	Embedded Operating Systems	5	Having completed this course, students should be able to - Account for advantages and disadvantages of Linux as operating system in embedded systems. - Describe the anatomy of a 32-bit embedded system. - Describe the features of a Beagle Bone system.	Having completed this course, students should be able to - Use basic Linux commands and utilities. - Select, install, configure and use tools needed for developing embedded systems. - Execute a firmware upgrade on a Beagle Bone system. - Install and configure “off the shelf” software in Linux.	Having completed this course, students should be able to - Implement shell scripts in BASH - Design and implement IoT-devices, based on a 32-bit MCU platform with Linux

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			- Describe the boot process of a 32-bit ARM based Linux system. - Explain Pulse Width Modulation. - Explain I2C communication bus technology. - Explain the structure of Linux file system and access permissions. - Explain how to connect and read input from sensors in an embedded Linux environment. - Explain how to connect and control actuators in an embedded Linux environment.	- Use the GPIO structure in Linux to interface sensors and actuators. - Use Pulse Width Modulation for Control of servo motors, and LED light intensity. - Implement BASH scripts to control simple GPIO devices. - Implement simple hardware circuits for measurement and control. - Use appropriate programming language to implement web-based user interface.	
IT-ERP1	ERP systems SAP ABAP/4 Programming	5	Having completed this course, students will be able to: - Understand the ABAP Workbench. - Create basic ABAP Programs. - Understand the control flow and structures in ABAP	Having completed this course, students will be able to: - Create Database with domains, data elements and tables - Retrieve Data from the Database with open sql. - Develop a simple ABAP Programs with modularization. - Develop a DYNPRO with navigation (CRUD-functionality) - Develop reports with selection screen, alv-list, etc.	Having completed this course, students will be able to: - Use the fundamental concepts of the ABAP programming Language - Create simple application programs with user dialogs and database connections. - Trace the flow of a program and troubleshoot simple problems. - Describe change management for new systems. - Use ABAP Workbench and basic ABAP language elements. - Explain the relationship and difference between the classical procedural programming model and the object-oriented programming model in ABAP/4. - Apply screen flow logic and working with external data. - Apply different tools and techniques available to implement dataflow in an ABAP program with database. - Design and implement an object-oriented SAP application with a database and ALV Grid.
IT-GMD1	Game Development	5	After successfully completing the course, the student will have gained knowledge about: - General game development principles, tools, patterns and best practices - Game engines and real-time development platforms - Unity, a cross-platform game engine	After successfully completing the course, the student will have acquired skills in: - Navigating Unity and making use of its various features - Creating C# scripts to modify game behaviour - Structuring game development projects - Deploying applications on various hardware	After successfully completing the course, the student will have acquired competences in developing industry standard interactive experiences using Unity. The student will be able to possess the developer position within a multidisciplinary game development pipeline, identifying and

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			- Code structure in a framework based on a loose implementation of the ECS pattern - Basic game design principles - The game development production pipeline - An engineer's role in game development - Intermediate programming concepts such as coroutines, delegates and events - The animator state machine - Digital audio - Various assets in game development - Interactive 3D rendering - The .NET framework	- Navigating the Unity and C# documentation - Utilizing game design theory to conceptualize games - Importing and working with various assets from other game development professions including 3D models and animations - Working with materials, shaders and textures - Handling physics in games - Creating and manipulating animations for characters using state-based machines - Creating responsive user interfaces for games - Working with digital audio in real-time engines - Optimization utilizing the profiler of Unity - Utilizing the SOLID design principles in a script-based environment	executing on the technical requirements of the developed product. The student will also have a solid foundation to further professional skills in game development independently.
IT-HWP1	Hardware Oriented programming	5	Having completed this course, students should be able to - Seek information in datasheets for electronic components - Describe the difference between polling and interrupt-based drivers - Describe layered software design and Hardware Abstraction Layer - Explain the Interrupt system in a microcontroller - Explain the concept of Pulse Width Modulation - Explain Timer/Counters and give examples of their use - Explain how analogue signals are sampled and quantified.	Having completed this course, students should be able to - Implement low-level drivers for digital I/O-Ports - Implement low-level drivers for analogue sensors - Implement low-level drivers for analogue actuators.	Having completed this course, students should be able to - Design a Hardware abstraction Layer - Implement low-level drivers for 8-bit microcontrollers.
IT-MAL2	Introduction to Machine Learning and AI	5	After having completed the course, the student will have gained knowledge about algorithms, methods, techniques, tools, and applications within the following: - Different types of neural networks, e.g., feed-forward, convolutional and recurrent neural networks. - Predictive methods, e.g., image classification and speech recognition.	Upon completion of this course, students should be able to: - Understand and apply a range of deep learning methods for AI. - Implement and fine-tune deep learning models in a programming language. - Apply ethical considerations when developing AI systems.	Upon completion of this course, the goal is that the students have acquired the competences to: - Make informed choices about the use of deep learning methods. - Communicate and discuss the theory, tools and techniques of deep learning and artificial intelligence.

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			- Generative methods, e.g., generative adversarial networks (GANs) and generative pre-trained transformers (GPTs). - Reinforcement methods, e.g. game AI.		- Discuss, address and reflect upon ethical aspects of using artificial intelligence.
IT-NSQ1	No-SQL versus relational databases	5	The student should be able to - describe document-based and graph databases - explain updating and querying in different database paradigms - explain updating and querying in GraphQL - explain schemas and constraints in non-relational databases • - compare relational and different non-relational approaches to database design.	The student should be able to apply - modelling techniques in document-based and graph databases - schemas and constraints to enforce designs in a no-SQL database - APIs and languages to maintain and query databases • - setting up No-SQL databases in the cloud	At the end of the course, the students should be able to - make an informed choice of database management system - design and create a data model in the chosen database system • - set up a cloud environment to use for the data model
IT-PCL1	Programming Concepts and Languages	5	Having successfully completed this course, the student should be able to: - describe the key concepts and have a basic understanding of different programming paradigms and languages. - understand and use the functional paradigm. - develop small and medium size programs/apps using F# and Python programming languages. - Understand and use the Python language with various middleware such as Django, Flask, and RabbitMQ	Having successfully completed this course, the student should be able to: - use the different programming paradigms to solve a particular programming problem. - Use one of the Web Frameworks to develop a distributed application.	
IT-PRM1	Project Management	5	The students will be able to: - Apply the planning process method to a complex project - Describe and explain what it takes to manage and run a complex project	The students will be able to use the methodology and tools for… - Estimating Project Time and Costs - Planning a Project - Using Risk Management - Conducting Team Management - Completing a project	During the course the students will work with analysis of a real time project and by applying acquired theoretical knowledge being able to outline used methods and tools including: - Project Description / Scope - Project plan - Project organization - Risk Analysis - Communication plan based on stakeholder analysis All leading to successfully managing and controlling a project.
IT-RTP1	Real-Time Programming,	5	Having completed this course, the student has gained knowledge in the below areas. Specifically, the student is able to:	Having completed this course, the student should be able to:	Having completed this course, students should be able to:

Code	Title	ECTS-points	Knowledge	Skills	Competencies
	Interfacing and Electronics		- Understand the basic concepts of real-time programming - Explain issues like deadlocks, priority inversion etc.	- Write functioning real-time programs in C using FreeRTOS - Analyse a simple real-time design for schedulability, deadlocks, utilization etc.	- master and use simple real-time operating systems - be able to analyse/design/describe and construct real-time programs - understand timers and clocks, and how they are used in real-time programming - understand synchronization avoiding dead-locks and priority inversion - understand memory management, resource sharing and control - be able to design and construct real-time systems using FreeRTOS and C-programming - understand low-level protocols, CRC etc.
IT-SCP1	IT Security and Cryptography in Practice	5	After successfully completing the course, the student will have gained - an understanding of the cross-disciplinary nature of cyber security, and the complexities, challenges and wider implications of the contexts in which cyber security problems occur in the workplace. - knowledge about several key implementations of cryptography and other IT-security related issues.	After successfully completing the course, the student will be able to - Draw on and apply relevant IT security approaches, tools and frameworks for IT security enquiry to different settings in real world situations. - Frame and address IT security problems, questions and issues as a IT security project, being aware of the environment and context in which the problem exists.	After successfully completing the course, the student will have acquired competences in - Applying complex cryptographic primitives to real-world cases - Documenting and explaining an IT-security project clearly and unambiguously to peers - Reviewing, evaluating and reflecting upon knowledge, skills and practices in cyber security.
IT-SMP1	Stochastic Modelling and Processing	5	After successfully completing the course, the student will have gained knowledge about: - The main working tools and concepts of stochastic modelling - Probability theory and distributions - Confidence Intervals and Hypothesis Testing - Inferential statistics	After successfully completing the course, the student will be able to: - Apply results from basic probability theory including conditional probability - Use probability density and distributions functions of one and two variables - Account for random variables and random processes - Calculate and estimate errors and uncertainties.	After successfully completing the course, the student will have acquired competencies in: - Planning experiments and state hypothesis - Presenting statistical results from experiments - Modelling experimental data with regression - Analysing experimental results and test hypotheses
IT-UXU1	User Experience and Usability	5	Upon completion of this course, students will be able to: - Explain how sensed elements (colors, typography, imagery) evoke emotions - Explain emotional design using a 3-layer model - Recognize the significance of usability testing	Upon completion of this course, students will be able to: - Recognize common cognitive processes (e.g., perception, attention, memory). - Identify cognitive biases that impact user behavior. - Conduct usability tests with real users using both controlled and natural settings	Upon completion of this course, students will be able to: - Compare emotional design across different platforms (web, mobile, physical products) - Reflect on the impact of authentic storytelling in product narratives - Propose innovative emotional design solutions for specific contexts

Code	Title	ECTS-points	Knowledge	Skills	Competencies
			- Recognize the impact of cognitive load on usability.	- Analyse user feedback to improve a design using Instant Data Analysis. - Discuss the impact of emotional engagement on user behavior.	- Reflect on the emotional impact of design choices. - Compare different settings in usability testing - Evaluate the impact of design decisions on user satisfaction. - Discuss ethical considerations in UX design. - Apply advanced thinking and creativity in understanding the user's desire and behavior - Propose innovative solutions for enhancing social interactions in digital products.
IT-WEB3	Full-stack Development	5	The student should be able to - explain the elements of the TypeScript type system. - explain the function of TypeScript utility types. - explain the definition of functional programming. - explain relevant client programming design patterns. - explain mechanisms for rendering and re-rendering. - explain the components of reactive programming.	The student should be able to - apply object-oriented programming in TypeScript. - apply functional programming in TypeScript. - apply callbacks and higher-order functions in TypeScript. - apply at least two web client frameworks and at least two state management frameworks. - implement a server using TypeScript. - implement server-side rendering. - use reactive programming in the client. - argue for the choice of server-side and client-side rendering. - argue for the choice of state management techniques in web client.	At the end of the course, the students should be able to - design and implement a web application using one or more of the techniques and technologies taught in the course.
IT-XRD1	XR Development	5	After successfully completing the course, the student will have gained knowledge about: - XR, AR, VR, AV & MR terminology - State of the art and the evolution of AR & VR hardware - Use cases for AR and VR applications - Tracking technologies for XR - Display technologies for XR - ARCore and ARFoundation - XR Interaction Toolkit - The Unity XR tech stack & OpenXR - XR Interaction techniques - Rendering challenges in XR - Spatialized audio - OVR and similar integrations for hand tracking, locomotion and specialized interactions	After successfully completing the course, the student will have acquired the skills to: - Analyse and optimize an AR or VR development workflow - Compare and utilize various SDK offerings and libraries for XR development - Work with and reflect on the theory behind prominent challenges in the XR industry such as tracking, rendering, locomotion and input - Reflect on underlying sensor and display technologies for XR hardware - Classify XR applications and reflect on their use cases - Describe, compare and apply various interaction techniques in XR	After successfully completing the course, the student will have acquired competencies to develop marker based and markerless augmented reality applications and mobile virtual reality applications.