Bring ideas to life
VIA University College

Date: August 2021

Curriculum

Programme section

Bachelor of Engineering in Software Technology Engineering

Applicable to students enrolled in August 2021 and later.

Students enrolled before August 2021 will follow the structure and subjects of the curriculum 2017.
In case of a delay in a student's study programme, the design of a personal study plan may lead to a transition to this curriculum.

CONTENTS

1 Identity of the programme 4

2 Graduate profile for VIA Engineers 4

3 Teaching and working methods 5

4 Structure and content 5

5 Compulsory elements of the programme, 1st – 4th semester 8
5.1 1st semester: Single User Systems 8
5.2 2nd semester: Client/Server Systems 9
5.3 3rd semester: Heterogeneous Systems 10
5.4 4th semester: Internet of Things 11

6 Internship, 5th semester 12

7 6th and 7th semester 13
7.1 6th semester 13
7.2 7th semester 14

8 Electives 15

9 Workshops 16

10 Bachelorprojekt 16

11 Title and issue of degree 17

12 Appendix 1: Courses Software Technology Engineering Programme 18

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:

- Transpose technical research results as well as scientific and technical knowledge to 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 social, economic, environmental and occupational health 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 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 put into practice 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 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 held 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 tutor.
- 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 including exams is stated in the overview on page 7.

The official duration of the degree program 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 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 and in February 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 20 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 VIA's Studynet.

The programme is structured as illustrated below:

Semester	**Course**	**Course**	**Course**	**Course/Project**	**Project**	**Project**
Theme	**5 ECTS**	**5 ECTS**	**5 ECTS**	**5 ECTS**	**5 ECTS**	**5 ECTS**
7 semester ***Electives***	**Elective Course**	**Elective Course**	**Elective Course**	**BPR2** Bachelor Project		
6 semester ***Innovation and Electives***	**ADS1** Algorithms and Data Structures	**Elective Course**	**Elective Course**	**BPR1** Bachelor Project Preparation	**SEP6** Semester Project (Innovation)	
5 semester ***Internship***	**INP1** Internship					
4 semester ***Internet-of-Things***	**AND1** Android Development	**ESW1** Embedded Software	**DAI1** Data Analytics Infrastructure	**DOC1** DevOps and Cloud	**SEP4** Semester Project	
3 semester ***Heterogeneous Systems***	**SDJ3** Software Development with UML and Java	**CAO1** Computer Architecture and Organisation	**DNP1** .NET Programming	**NES1** Networking and Security	**SEP3** Semester Project	
2 semester ***Client/Server Systems***	**SDJ2** Software Development with UML and Java		**SWE1** Software Engineering	**DBS1** Database Systems	**SEP2** Semester Project	
1 semester ***Single User Systems***	**SDJ1** Software Development with UML and Java		**RWD1** Responsive Web Design	**DMA1** Discrete Mathematics and Algorithms	**SEP1** Semester Project	

Depending on the choice of electives, students may specialise in one of three areas:

- Internet of Things
- Interactive Media
- Data Engineering

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 as a number of introductory courses and a semester project in which the students will design, implement and document a single-user software system

The purpose of the courses, ECTS and assessment:

Software Development with UML and Java (SDJ1) – 10 ECTS	**Prerequisites for attending exam**	**Examination** (all re-exams may be oral)
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.	Attendance (≥ 75%) Mandatory course activities completed	Oral exam (30 minutes), weighing 80%. External assessment Mid-term test, weighing 20%
Responsive Web Design (RWD1) – 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.	Attendance (≥ 75%) 3 approved assignments	Written (2 hours) External assessment
Discrete Mathematics and Algorithms (DMA1) – 5 ECTS		
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.	Attendance (≥ 75%) Mandatory course activities completed	Written (4 hours) Internal assessment

Semester Project (SEP1) – 10 ECTS		
The purpose is to develop and document a single user system, demonstrate an acquisition of process skills, and to develop the student's basic skills and competences for the excellent performance of study and project related activities that are required in the process of working towards an engineering degree.	Reports handed in before deadline	Oral Group presentation approx. 20 minutes followed by a joint examination with a joint discussion and individual question rounds for approx. 20 minutes per student including voting. Internal assessment

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

ECTS credits: 30

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 purpose of the courses, ECTS and assessment:

Software Development with UML and Java (SDJ2) – 10 ECTS	**Prerequisites for attending exam**	**Examination** (all re-exams may be oral)
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%) Mandatory course activities completed	Oral (20 minutes) External assessment
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-lift problem. This involves requirement capturing (Use Cases and non-functional requirements), analysis, domain models, interaction diagrams, design classes, design patterns and test-descriptions etc.	Attendance (≥ 75%) Mandatory course activities completed	Oral exam Internal assessment
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).	Attendance (≥ 75%) Mandatory course activities completed	Written (4 hours) External assessment

Semester Project (SEP2) – 10 ECTS		
The purpose is to develop and document a client/server system as well as demonstrate the acquisition of process skills.	Project report handed in before deadline	Oral Group examination (5 min per student) Group examination (20 min per student) Internal assessment

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

ECTS credits: 30

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 purpose of the courses, ECTS and assessment:

Software Development with UML and Java (SDJ3) – 5 ECTS	**Prerequisites for attending exam**	**Examination** (all re-exams may be oral)
The students should be introduced to basic theory of distributed systems and be able to design and implement a distributed system.	Attendance (≥ 75%) Mandatory course activities completed	Oral exam Internal assessment
Computer Architecture and Organization (CAO1) – 5 ECTS		
The main purpose of the course is to gain a basic understanding of the organization and design of computers with a focus on the central processing unit (CPU) and the necessary logic involved in building a CPU.	Attendance (≥ 75%) Mandatory course activities completed	Written (2 hours) External assessment
.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.	Attendance (≥ 75%)	Written (3 hours) Internal assessment
Networking and Security (NES1) – 5 ECTS		
The main purpose of the course is to gain a basic understanding of computer networks, protocols and security technology in the Internet. A secondary purpose is to gain hands-on experience in network configuration of hosts and routers.	Attendance (≥ 75%)	Written (3 hours) Internal assessment
Semester Project (SEP3) – 10 ECTS		
The purpose is to develop and document a distributed, heterogeneous system and a network protocol, herein account for the security aspects of the system and the protocol.	Project report handed in before deadline	Oral Group examination (5 min per student) Individual examination (25 min per student) Internal assessment

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

ECTS credits: 30

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, an android-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 purpose of the courses, ECTS and assessment:

Android Development (AND1) – 5 ECTS	**Prerequisites for attending exam**	**Examination** (all re-exams may be oral)
The purpose of this course is to provide the student with the knowledge, skills and competencies needed to utilize the tools, principles, patterns and best practices of Android development.	Attendance (≥ 75%) Mandatory course activities completed	Written (3 hours) External assessment
Embedded Software (ESW1)		
The purpose is to qualify the student to apply basic concepts in real-time programming, and to implement real-time programs using the C-programming language on embedded micro-controllers and using interfaces (APIs) to a number of sensors and actuators.	Attendance (≥ 75%) Mandatory course activities completed	Written (3 hours) External assessment
Data Analytics Infrastructure (DAI1)		
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 from different sources, data modelling for analytics and basic visualization.	Attendance (≥ 75%) Mandatory course activities completed	Oral (20 minutes) External assessment
DevOps and Cloud (DOC1)		
The purpose of the course is to introduce students to a set of practices that help deliver software in a fast and reliable manner. By automating workflows and integrating the processes of software development and IT operations, the students will be able to create a consistent toolchain that supports planning, building and operating a software project.	Attendance (≥ 75%) Mandatory course activities completed	Written (3 hours) Internal assessment

Semester Project (SEP4) – 10 ECTS		
Conceive, design and implement a software solution including hardware sensors, an android-based user interface and a persistent multiuser backend infrastructure. The solution must contain self-constructed electronics, and make use of the Java, C# & C programming languages.	Project report handed in before deadline	The students will solely be evaluated on the project- and process report handed in, as well as a pre-recorded video presentation (max 30 minutes), that demonstrates the system and presents how the project has developed. Students must mark clearly what sections of the reports they have participated in and their git commit history must reflect their contribution to the implementation. Internal assessment

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.

ECTS credits: 30

6 Internship, 5th semester

IT-INP1

The internship comprises a semester of 30 ECTS and timewise is 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.

The internship is graded as passed/not passed. Internal assessment.

7 6th and 7th semester

The 6th and 7th semester consist of two mandatory courses (ADS1 and BPR1), elective courses, and projects.

The software student can select one of the three specialisations, but the student can also complete their degree without a specialisation. A specialisation consists of three elective courses (15 ECTS credits), and a substantial part of the bachelor project must be within the specialisation area. Apart from those courses, a number of relevant elective courses are offered in the Software Technology Engineering Programme. An overview of the individual elective courses is shown below in section 8. To obtain a specialisation, the following elective courses should be selected:

Interactive Media	At least 15 ECTS should be selected from the “Interactive Media” courses in section 8.
Internet of Things	BEL1, HWP1, and at least one additional “Internet of Things” course from section 8.
Data Engineering	BUI1 and at least 10 additional ECTS should be selected from the “Data Engineering” courses in section 8.

It is also possible to choose one elective course (5 ECTS) offered by other programmes at VIA. 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 (SEP6) within the area of innovation and entrepreneurship is completed in project groups of students from different engineering programs at VIA. In addition to SEP6, the student must pass two compulsory courses (ADS1 and BPR1) and two electives at 6th semester.

Algorithms and Data Structures (ADS1) – 5 ECTS	**Prerequisites for attending exam**	**Assessment** (all re-exams may be oral)
The purpose of the course is to qualify the student to design, implement and analyse different algorithms. Furthermore, the student will become acquainted with different advanced data structures.	None	Written (3 hours) External assessment
Bachelor Project Preparation (BPR1) – 5 ECTS		
The purpose of the course is to prepare the student for the Bachelor Project. In preparing the Bachelor Project students learn to recognize important sets of problems within the professional area, alternative solutions to them and the demands of companies and their environments. In the course the students are taught how to apply scientific knowledge and work methods to their own field in new and changing situations. They also learn to communicate orally and in writing on questions related to the area of research, as well as methods for collecting data and testing their solutions.	Mandatory course activities completed	Approved/Not approved

Each bachelor project group must consist of 2-3 students.		
Electives		
For further details see section 8	For further details see section 8	For further details see section 8
Semester Project (SEP6) – 10 ECTS		
Semester Project – Innovation project execution in a cross professional environment	Mandatory course activities completed	Oral examination External assessment

7.2 7th semester

The programme is concluded with a bachelor project (BPR2) which constitutes 15 ECTS credits of the programme. The bachelor project is initiated on 6th semester (BPR1, 5 ects) where among other aspects the project description is completed.

The bachelor project must demonstrate individual self-critical reflection within the chosen area and must document the student's ability to apply engineering theories and methods. In addition, the bachelor project must reflect the student's ability to communicate professionally.

The requirement for attending the bachelor project (BPR2) examination is that the student has passed all courses in the 1st - 7th semester (courses totalling 195 ECTS, including the 30 ECTS internship). The bachelor project is prepared in groups of two or three persons.

Electives	**Prerequisites for attending exam**	**Assessment** (all re-exams may be oral)
For further details see section 8	For further details see section 8	For further details see section 8
Bachelor Project (BPR2) – 15 ECTS		
The purpose of the Bachelor Project 2 is to evolve the student’s ability to solve a relevant software engineering problem and document the solution. 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. Each bachelor project group must consist of 2-3 students.	Project report handed in before deadline	Oral Group presentation (20 min) Individual examination (20 min per student) External assessment

8 Electives

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

Courses (5 ECTS)	Specialisation	Prerequisites for attending exam	Assessment (all re-exams may be oral)
Programming Concepts and Languages (PCL1)	None	Mandatory course activities completed	Written (3 hours) Internal assessment
Applied Linear Algebra (ALI1)	None	None	Written (3 hours) Internal assessment
Process Management for Software Engineering (PME1)	None	Mandatory course activities completed	Oral (20 minutes) Internal assessment Final grade: 50% course work, 50% exam
Compiler Construction (CMC1)	None	Mandatory course activities completed	Oral (20 minutes) Internal assessment
ERP Systems SAP ABAP/4 Programming (ERP1)	None	Mandatory course activities completed	Oral (20 minutes) Internal assessment Final grade: 50% course work, 50% exam
Advanced .NET Programming (DNP2)	Interactive media	Mandatory course activities completed	Oral (20 minutes) Internal assessment Final grade: 50% course work, 50% exam
Single-page Web Applications (SWA1)	Interactive media	Mandatory course activities completed	Oral (20 minutes) Internal assessment
Interaction Design (IDX1)	Interactive media	Mandatory course activities completed	Oral (20 minutes) Internal assessment Final grade: 50% course work, 50% exam
Game Development (GMD1)	Interactive media	Mandatory course activities completed	Oral (20 minutes) Internal assessment
Digital Multi Media (DIM1)	Interactive media	Mandatory course activities completed	Oral (20 minutes) Internal assessment Final grade: 50% course work, 50% exam
Mixed Reality (MIX1)	Interactive media	Mandatory course activities completed	Oral (20 minutes) Internal assessment
Basic Electronics (BEL1)	Internet of Things	Mandatory course activities completed	Oral (20 minutes) Internal assessment
Digital Signal Processing (DSP1)	Internet of Things	Mandatory course activities completed	Oral (20 minutes) Internal assessment
Embedded Operating Systems (EOS1)	Internet of Things	Mandatory course activities completed	Oral (20 minutes) Internal assessment
Real-Time Programming (RTP1)	Internet of Things	Mandatory course activities completed	Oral (20 minutes) Internal assessment
Internet-of-Things WAN's (LWA1)	Internet of Things	Mandatory course activities completed	Oral (20 minutes) Internal assessment

Hardware Oriented Programming (HWP1)	Internet of Things	Mandatory course activities completed	Oral (20 minutes) Internal assessment Final grade: 25% course work, 75% exam
Machine Learning (MAL1)	Data Engineering	Mandatory course activities completed	Oral (group exam (30 minutes), individual exam (15 minutes)) Internal assessment Final grade: 50% group exam, 50% individual exam
Stochastic Processes and Modelling (SMP1)	Data Engineering	None	Written (3 hours) Internal assessment
Business Intelligence (BUI1)	Data Engineering	Mandatory course activities completed	Oral (20 minutes) Internal assessment
No-SQL versus relational databases (NSQ1)	Data Engineering	Mandatory course activities completed	Oral (20 minutes) Internal assessment Final grade: 50% course work, 50% exam

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

9 Workshops

The program covers two workshops WS1 (1st semester) and WS2 (2nd semester) and both workshops are related to the curriculum in respectively SDJ1 and SDJ2:

WS1:
Initially, a test is completed which result determines whether the student must take part of WS1 or not (the test will take place within the first five weeks of 1st semester).

WS2:
Students who achieve 10 or 12 at the SDJ1-examination do not have to follow WS2. All other students must follow WS2.

In order to pass WS1 and WS2 a minimum of 75% attendance is required.

10 Bachelor Project

IT-BPR1
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 examination. 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.

A condition for being able to commence the bachelor project is the student being assessed as being likely ready for the examination, as BPR2 must be the last examination of the study.

As a rule, the bachelor project is prepared in groups of 2-3 people.

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 examination/group examination with individual assessment in accordance with the programme’s overall goals as described in Section 1 of the Curriculum. The basis for examination 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.

Examinations can take place at the earliest when all the other examinations of the programme, including internships, have been passed. The examination is assessed according to the 7-point scale and with the participation of an external assessment.

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 Engineering.

It is also possible to obtain the following special designations:

- Interactive Media
- Internet of Things
- 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, proof of having passed study units will be issued.

12 Appendix 1: Courses Software Technology Engineering Programme

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
IT-DMA1	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	Prerequisites for exam: If the following requirements are not met, the student will not qualify for the exam: i. The student must have an attendance of at least 75%. ii. The student must hand in 8 mandatory assignments. Type of Examination: The course is evaluated based on a 4 hour written test. The test is completed in the Flowlock browser in Wiseflow. The final grade will reflect an overall assessment of the mandatory assignments and the written test. The final test must, however, be awarded a passing grade in order for the final grade to be a passing grade. Internal assessment. 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: Completed as the ordinary exam, but may be decided changed to oral exam.
IT-RWD1	Responsive Web Design	5	Having completed this course, students will have the knowledge to: • Describe the different file formats	Having completed this course, students will have the skills to: • Create web sites using Hyper Text Markup Language (HTML5).	Having completed this course, students will be able to: • Design and implement platform independent web applications.	Prerequisites for exam: If the following requirements are not met, the student will not qualify for the exam:

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			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.	• 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 XMLHttpRequest to read content from an external source and integrate this content into a website. • Select HTML elements and apply jQuery animations to the selected elements to make websites interactive.		The student must have an attendance of at least 75%. Mandatory assignments handed in before deadline and accepted. Exam type: Digital written examination duration of 2 hours (2 parts): Part 1: Multiple choice questions 30 minutes Part 2: Short answer questions 90 minutes (explaining and writing code) 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. Re-exam: Completed as the ordinary exam, but may be decided changed to oral exam.
IT-SDJ1	Software Development with UML and Java	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. Im-	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.	Prerequisites for exam: Mandatory course activities completed. The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of Examination: Individual oral examination where the student will pick an unfamiliar programming exercise at random. The student must explain the UML

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
				plement 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.		involved and demonstrate how to perform the programming task using a laptop. The time allotted for the examination is 30 minutes including assessment. Exam is with an external assessment. Tools allowed: All. Re-exam: Same as ordinary exam.
IT-SEP1	Semester Project: Single User System	10	Course-related learning outcome The student will use the knowledge acquired in SDJ1, RWD1 and DMA1. Effective teams Account for covered theories on group dynamics, team work and conflict resolution Own learning process Refer to covered theories on learning, motivation, feedback and study techniques Project framework Identify relevant knowledge in connection with written academic and technical communication including report structure, reference handling and source management Identify and apply presentation techniques relevant for the target audience Problem Based Learning Explain basic elements within Problem Based Learning Identify relevant problem statements and identify specific demands for a problem statement Project management Identify relevant project management methods, including planning,	Course-related learning outcome Explain the Waterfall method as a software development process Derive requirements Apply use case modelling and draw activity diagrams Draw a domain model Construct 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 how to use your system in a user guide Explain and discuss the time and space usage of algorithms and data structures Effective teams Formulate and enforce a group contract together with the group Establish and be part of a cooperation with the project group and the supervisor Own learning process Apply learning theories and motivational theories in connection with own learning process as well as	Professional learning outcome Demonstrate the connection between the different stages in software development Evaluate the performance of selected parts of their program in terms of time and space complexity using the Big O notation Effective teams Describe and reflect on the project group's cooperation – including own effort – to define areas for improvement in future projects Own learning process Reflect on own ability to learn from different teaching and study activities including the project group's work Problem Based Learning Take responsibility for the student directed part of the semester project	Prerequisites: Project report and process report must be submitted before the deadline set by the supervisors. Type of examination: Group exam with individual assessment. Group presentation approx. 20 minutes followed by a joint examination 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 examination. Internal assessment Allowed tools: All Re-exams: Students who failed a semester project in January or June must attend an information meeting on the last Friday in June. At this meeting, the students will get information on specific deadlines as well as the process of re-exam.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			meeting management, risk assessment and quality assurance	give and receive feedback Project framework Have a critical approach to sources, use references, apply proper reference management including comply with the rules for plagiarism Communicate the results of the project work and the learning process of the project group in a structured way using technical terminology both in writing, graphically and orally Communicate successfully in writing and orally to different target groups Problem Based Learning Define a problem statement, describe different solutions and account for proposed solution Project management Account for choice and application of tools and methods for project management in order to reach specific goals in the project work.		They will form new groups, if possible in relation to the number of failed students at the individual semesters. 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 in September.
IT-DBS1	Database Systems	5	Having completed this course, students will be able to: explain the relationship between relational algebra and SQL. explain the relational model. explain the 3 normal forms. explain keys in relational databases. explain joins. explain transactions	Having completed this course, students will be able to create ER-Models with UML 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	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	Criteria to qualify for the exam: Course assignments handed in before deadline The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Examination Duration: 4 hours Digital written examination (2 parts): Part 1: Multiple choice and written answers; 1 hour without aids Part 2: Design and implementation; 3 hours with all aids, including

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
						internet connection External assessment Re-exam: Completed as the ordinary exam, but may be decided changed to oral exam.
IT-SDJ2	Software Development with UML and Java 2	10	The student should be able to understand: System architecture Different methods for testing Concurrency System deployment 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. Create jar files 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	Criteria to qualify for the exam: • Course assignments handed in before deadline • The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of Examination: Individual oral examination 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.
IT-SEP2	Semester Project: Client/Server System	10	The student will be able to: Describe the relation between design and test in relation to a software system Define and use group roles when using Scrum to control a software project Describe how to deploy a software system	The student will be able to: Use an iterative system development method when executing a software project Execute effective literature searching, in order to judge validity and reliability Document the Analysis and Design of a software system using UML Capture and formulate the requirements, both functional and non-functional, of a software system Write a project description for a software project Design and describe object-oriented models Design a normalized relational database for data persistence, based on the preliminary	The student will be able to: Capture requirements, analyse, design, implement and test a client/server system using UML, Java and Silcrete and ague for a design that supports team collaboration of implementation and tests Execute a software project with a clear connection between requirements, analysis, design, implementation, test and documentation Create a small, robust, concurrent client/server system with the proper selection of design patterns and database-persistence to achieve a maintainable and flexible	Permit criteria for attending examination: •The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. •Mandatory assignments and reports handed in before deadline and accepted. •The uploaded reports must comply with the hand in criteria as stated in the Support Document: Formalities Criteria for upload of SEP.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
				analysis Use SCRUM to control the development process of a software project Use Unified Process as a system development method Display considerable skills for presentation, both written and oral Construct a project report in a well-structured manner, using provided templates Describe a project execution in a process report Conduct and execute a software project in collaboration with group members Perform unit testing and use case testing based on the requirements and code of a software system Implement a client-server software system in Java according to the system’s design Implement a relational database as part of a client-server system according to the system’s design Apply theory of database normalization that will result in a relational database on 3rd normal form	architecture Choose between, argue for, and evaluate various technical solutions for implementing client/server systems Describe and reflect on the development process and project work in a Process Report Black- and Whitebox testing of a software system using relevant testing techniques Apply relevant design principles including SOLID principles so that the resulting software system is maintainable, flexible, and robust Reflect upon chosen technical solutions of the software system Discuss their choice of design patterns	Found under Study Material -> Project Guidance -> ICT Type of Examination: Oral Examination Group presentation - 5 minutes per person Group examination - 20 minutes/student SWE is evaluated together with SEP2 project. The SEP2 project and the exam must demonstrate understanding of SWE1 topics and their use in practice During the SEP2 exam, specific SWE1 questions will be asked, which must be answered satisfactorily to pass SEP2 Allowed tools: Laptops, notes Internal assessment. Re-examination Students who failed a semester project in January or June must attend an information meeting on the last Friday in June. At this meeting, the students will get information on specific deadlines as well as the process of re-exam. They will form new groups, if possible, in relation to the number of failed students at the individual semesters. Based on the feedback, the students have received after the ordinary exam, they must prepare a new project. 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

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
						takes place in August/September.
IT-SWE1	Software Engineering	5	The student should be able to account for: Abstraction UMLS.O.L.I.D Design principles Unified Process SCRUM Design principles Architecture design Requirement capturing Analysis vs. Design models The difference between software development and coding How to follow a test description	The student should exemplify: Analysis of a problem and document the analyse- and design-process in UML Practical use of UMLPractical use of Unified Process UML to document requirements, analysis and design artefacts Unified Process in combination with agile software development SCRUM together with Unified Process A domain model from a problem description and requirement specification and the elements in the model A design model and understand the elements within it The S.O.L.I.D principles on a design model Design for test Test descriptions Architectural design models	The student should be able to: Analyse a problem– what is the problem to solve? Identify a problem and derive a requirement specification with Use Cases and non-functional requirements Plan tests Analyse and design a project to be implemented in teams with many participants	Permit criteria's for attending examination: Mandatory course activities completed before deadline. The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of Examination: IF you are full degree Software Engineering student: SWE 1 is evaluated together with SEP 2 project The SEP 2 project and the exam must demonstrate understanding of SWE 1 skills and competencies and their use in practice During the SEP 2 exam, specific SWE 1 questions will be asked, which must be answered satisfactorily to pass SEP 2 ELSE Individual oral examination without preparation based upon course assignment(s) Individual oral examination based upon a subject found by draw No preparation Oral examination where: The student will pick a familiar question at random. The student must explain the concepts and theories about a subject from the course (50%). The student will present a prepared presentation about the course assignment(s) (50%). Allowed tools: All. ENDIF

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
						Internal assessment.
IT-CAO1	Computer Architecture and Organization	5	Having completed this course, the student has gained knowledge in the below areas. Specifically, 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.)Describe instruction set layout and identify memory architectures and addressing modes.	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.	Permit criteria for attending examination: •In order to qualify for the examination, two course assignments must be handed in before deadline and approved. •The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of Examination: Written examination. Duration: 2 hours. External assessment. Examination is digital, and the students are required to bring a laptop that are tested and ready for use with Wiseflow and FlowLock. Allowed tools: Course literature according to the course description Personal notes.
IT-DNP1	.NET Programming	5	The student will be able to: Identify the C# programming language Describe the fundamentals of .NET development and the common type system Describe how RESTful web services are implemented in a distributed system Identify and describe .NET technologies relevant to web application development Describe how data can be accessed through object-relational mapping Define basic authentication and authorization	The student will be able to: Write and debug C# code Navigate and use the managed .NET API Create and consume class libraries Implement console applications, web applications and web services with Server-side and client-side C#-programming Data persistence using object-relational mapping User management, including authentication and authorization Consume and expose RESTful web services Deploy .NET applications Use a command-line interface (CLI) toolchain Compare object-relational mapping to traditional data access tech-	The student will be able to: Analyse and evaluate the relevance of .NET technologies when designing software applications Develop .NET applications and services as a part of a distributed system, herein account for communication protocols used	Permit criteria for attending examination: •Mandatory course assignment handed in before deadline and accepted. •The student must have an attendance of at least 75% to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of Examination: Written examination Duration: 3 hours Programming exercises; with all aids, including internet connection Internal assessment Re-exam:

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
				niques Explain the difference between client-side and server-side programming		Completed as the ordinary exam, but may be decided changed to oral exam.
IT-NES1	Networking and Security	5	The students can define the term “protocol” account for layered abstractions in protocol stacks can explain the Internet’s naming system explain addressing in the Internet identify common threats in the Internet describe common access control systems, e.g. packet filter, proxy, etc. Use common encryption technologies explain how network properties affect application performance.	The student can analyse network traffic using packet sniffer software compare and contrast different encryption technologies discuss how confidentiality, integrity and availability can be accomplished using encryption technology create simple cryptographic keys for use in network settings.	The student can create and maintain a simple LAN with several computers and one router create application layer protocols for distributed systems identify security threats and propose mechanisms to mitigate these threats.	Permit criteria for attending examination: •The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of Examination: Written examination. Duration: 3 hours. Internal assessment. Allowed tools: Course literature according to the course description Personal notes on paper Laptop (no access to local files or general internet)
IT-SDJ3	Software Development with UML and Java 3	5	The students will be able to describe various distributed system types (e.g. client/server, peer-to-peer)explain the 3-tier architecture explain various distributed communication methods list examples of distributed algorithms	The students will be able to use various distributed communication methods (e.g. web services, message queues)argue the choice of middleware for a given distributed system	The students will be able to design the architecture of a distributed system using the 3-tier model design and implement a distributed system on different platforms using various middleware	Permit criteria for attending examination: •At least 75 % attendance Type of Examination: Oral Examination The exam is a joint exam with IT-SEP3. Group presentation followed by individual examination. Group presentation of the IT-SEP3 project - 5 minutes per person. Individual examination - 25 minutes including examination in IT-SEP3 and IT-SDJ3. Internal assessment. Allowed tools: All
IT-SEP3	Semester Project:	10		The student should be able to Formulate a project description for a	The student should be able to Analyse, design, implement and test a	Permit criteria for attending examination:

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
	Heterogeneous System			software system using a distributed architecture Seek appropriate literature Apply architectural design patterns Implement heterogeneous systems using sockets and web services Design application layer protocols Use TLS/SSL to enhance security Analyse the security risks of a distributed system Use a defined methodology to control the development process Apply a system development method to produce the documentation of the system Use a version control system to manage versions	distributed system using UML and at least 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 Plan a development project for a distributed system Describe and reflect on the development process and project work.	•Group reports including student's name handed in before deadline. Type of Examination: Oral Examination The examination is a joint exam with IT-SDJ3. Group presentation followed by individual examination. Group presentation of the project - 5 minutes per person Individual examination - 25 minutes including examination in IT-SDJ3. Internal assessment. Allowed tools: All Re-examination: Students who failed a semester project in January or June must attend an information meeting on the last Friday in June. At this meeting, the students will get information on specific deadlines as well as the process of re-exam. They will form new groups, if possible, in relation to the number of failed students at the individual semesters. Based on the feedback, the students have received after the ordinary exam, they must prepare a new project. 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 in August/September.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
IT-AND1	Android Development	5	After successfully completing the course, the student will have gained knowledge about: General mobile development principles, tools, patterns and best practices The Android framework and development environment User interface definitions and material design Application components including Activities and Fragments Application communication related to Intents & intent filters Resources for externalization, localization and visualization Data persistence though Shared Preferences and SQLiteNetworking including JSON parsing, threads & http requests External services including Google Maps and Google Firebase Alternatives to the Android platform	After successfully completing the course, the student will have acquired the skills to: Navigate Android Studio and make use of its various features Deploy applications on mobile devices and emulators Use XML to create responsive user interfaces Implement with various UI-components Navigate and use the Android Developer documentation Optimize applications to run on mobile devices Send data in and between applications Utilize the Java language to implement application logic Incorporate external services to extend app functionality Properly handle application resources Make use of the various data storing methods on Android Construct Android applications using the MVVM architecture Use Google services such as Firebase and Google Maps Test applications with unit tests Publish applications on the Google Play Store	After successfully completing the course, the student will have acquired competencies in developing industry standard mobile and ubiquitous computing applications. The student will be able to partake the developer role within a multidisciplinary mobile development team, identifying and executing on the technical requirements of the developed product. The student will also have a solid foundation to further professional skills in mobile development independently.	Permit criteria for attending examination Mandatory assignments handed in before deadline and accepted. Type of Examination: Three hour written exam. External assessment. Re-exam: Completed as an ordinary exam, but can be completed as an oral exam.
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: Use scenarios for analytical data processing, 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	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	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.	Permit criteria for attending examination: •Mandatory course activities completed •Course assignment handed in before deadline Type of Examination: Oral examination Individual oral examination without preparation based upon course assignment(s) Oral examination, covering mandatory course work and theory covered in the course.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			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, validating and cleansing data, integration and publishing of data.	data Design, implement and test basic visualizations and graphs of data and analysis results.		Duration (grading included) app. 20 min/ 5 ECTS. External assessment Allowed Tools: All
IT-DOC1	DevOps and Cloud	5		The purpose of the course is to introduce students to a set of practices that help deliver software in a fast and reliable manner. By automating workflows and integrating the processes of software development and IT operations, the students will be able to create a consistent toolchain that supports planning, building and operating a software project.		Permit criteria for attending examination: •The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of Examination: Written examination. Duration: 3 hours. Internal assessment.
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 Basic concepts of programming with a Real-time Operating System (RTOS)Issues like deadlocks etc. Real-time C-programs for embedded Micro Controller Units 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 real-time programs in C using FreeRTOS Implement programs in C using different C API's and libraries for hardware drivers etc. Document C source code with Doxygen	The student should be able to: Design and construct real-time systems using FreeRTOS and C-programming Construct real-time programs Apply FreeRTOS timers in real-time programs in C Apply synchronization and avoid dead-locks Apply memory management, resource sharing and control	Permit criteria's for attending examination: •Mandatory course activities completed Type of Examination: Three hour written exam. External assessment. Allowed tools: Laptop
IT-SEP4	Semester Project 4	10	After successfully completing the course, the student will have gained knowledge about: Implementing Real-time operating systems (RTOS)Using LoRaWAN Implementing programs in C Constructing Android applications Applying Data warehouse modelling/	After successfully completing the course, the student will have acquired the skills to: Implement complete Android applications using a modern Android development environment Use external libraries to send and retrieve data from a web- to an Android application	After successfully completing the course, the student will have acquired competencies to: Analyse and design complete solutions comprised of both hardware and software Decide on appropriate quality assuring methods for a	Permit criteria for attending examination: Project- and process report handed in before deadline Type of Examination: The students will solely be evaluated on the project- and process report handed in, as well as a pre-

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			dimensional modelling Using Extract, Clean-up, Transform and Load data flows	Setup and maintain a build server for a larger software project Setup and maintain automated regression testing Implement RTOS-based applications in C Apply knowledge of dimensional database modelling to design databases optimized for querying Plan, design and implement Extract, Clean-up, Transform and Load data flows from multiple sources into a data warehouse Design and implement analyses based on the data warehouse	given software development project Implement full-scale Internet-of-Things solution Design and implement a data warehouse solution Develop industry standard mobile applications Conduct projects in multidisciplinary teams	recorded video presentation (max 30 minutes), that demonstrates the system and presents how the project has developed. Students must mark clearly what sections of the reports they have participated in and their git commit history must reflect their contribution to the implementation. Internal assessment <u>Allowed tools</u>: All <u>Re-examination:</u> Students who failed a semester project in January or June must attend an information meeting on the last Friday in June. At this meeting, the students will get information on specific deadlines as well as the process of re-exam. They will form new groups, if possible in relation to the number of failed students at the individual semesters. 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 in late August.
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	The student must: • be able to apply the methodologies and tools of one or more fields of study and to apply skills related	The student must: • be able to handle complex and development oriented situations in study or work contexts	In order to get an internship evaluated, the student must fulfil the following requirements concerning mandatory assignments:

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			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.	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.	• 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	• Expected outcome/specific learnings targets for the internship position • Company presentation • Logbook • Main academic assignment(s) • Final reflections • Participation in workshop for coming interns Internal assessment.
IT-ADS1	Algorithms and Data Structures	5	Know different linear data structures (sets, maps, lists and stacks)Know different algorithm types and templates Know the concept of Abstract Data Types Know different Sorting and searching algorithms Know different nonlinear data types (Trees, Heaps and Graphs)	Be able to analyse algorithms using the Big-O notation Be able to design and implement algorithms and data structures in an object oriented language	Be able to use algorithms and data structures to solve specific non-trivial problems. Be able to make good choices of data structures for a specific problem Be able to design and implement effective programs Be able to analyse and improve existing programs Be able to analyse algorithms using Big-O notation	Permit criteria for attending examination: •Mandatory course activities completed Type of Examination: Written examination Duration: 3 hours External assessment Allowed tools: Course literature according to the course description Personal notes Re-exam: Completed as the ordinary exam, but may be decided changed to oral exam.
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	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	The final exam is a 3 hour written exam and takes place at Campus Horsens. All supplementary materials and aids are allowed, e.g. using a computer as a reference work. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Internet access is not allowed.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
				Express mathematically correct arguments Use mathematical terminology and symbol language	linear algebra in the solution of engineering problems	Internal assessment. The exam must be completed in the Jupyter Notebook environment and the answers must be submitted in Wiseflow. The re-exam may be held as an oral examination.
IT-BEL1	Basic Electronics	5	Having completed this course, students should have understanding of: Statistics, Observation variance and error Ohm's law and Kirchhoff' law used on small electronic circuits Theory of basic analogue electronic components (resistor, capacitor)Operation Amplifiers, Instrument Amplifier, Diode, transistor (NPN, PNP)Cooling of electronic components (Heatsink, Compound)Simulation and practical build in Lab of small electronic circuits Low pass, Band pass, High pass, Butterworth filter properties Strain Gauges used in Wheatstone Bridge	Having completed this course, students should be able to: Simulate analogue electronic circuits using simulation software Construct active filters with desired property for specific application Construct electric car model on breadboard Build and test prototype circuits Perform measurements on electronic circuits, using Digital Multi Meter and Pico-scope	Having completed this course, students should be able to: Design simple electronic circuits for measurement systems using amplifiers and filters Analyse experiment results, using statistical calculations and methods Write reports to document engineering experiments	Individual oral examination based upon a subject found by draw. No preparation Duration 0.3 hours Allowed tools: All Internal assessment The course must be passed before "Practical Placement".
IT-BPR1	Bachelor Project Preparation	5	After having completed this course, the student should be able to: Explain the concept of plagiarism and how to avoid it Identify a problem and a problem domain Evaluate teamwork and team dynamics Select relevant methods for developing a project	After having completed this course, the student should be able to: Perform information search and retrieval Apply selected theories on their own group and identify potential challenges Describe a proposed problem to solve, as well as its context Plan and delimit a software development project Find relevant guidelines and templates	After having completed this course, the student should be able to: Describe a larger Software Engineering project in a Project Description Apply the preliminary steps in a system development process Define clear and concise requirements using a selected standard Demonstrate the ability to work coherently in a group	Permit criteria for attending examination: •Mandatory course activities completed Type of Examination: The basis of the evaluation is three mandatory assignments: •Group Description and Group Contract •Project Description •Requirements/User Stories
IT-BUI1	Business Intelligence	5	Students will obtain knowledge about understanding, reading, and displaying data from a dimensional model, such as a star scheme or data cube.	Data migration using integration services Designing paginated reports in Reporting services Scheduled jobs in SQL server Creating analyses in Power BI Creating cubes in Analysis services		Oral examination based on the course assignment. Approximately 20 minutes incl. discussion of examinee's performance, without preparation. Internal assessment.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
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 organization 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	The exam is an individual 20 minutes oral exam. The student draws a question from the theory and answers it based on the produced course project. Allowed tools: All Internal exam.
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	Oral examination. Approximately 20 minutes (including discussion of examinee's performance). The exam is in two parts. First part is a presentation and discussion of selected parts of the course work. Second part is based on a drawn question from the course syllabus. Internal assessment.
IT-DNP2	Advanced .NET Programming	5			Having completed this course, students should be able to: Master advanced features of C# language Having the advanced knowledge of the .NET framework and be able to choose the best technology for different problems Having the knowledge to use the different .NET technologies in projects.	Oral examination. Approximately 20 minutes (including discussion of examinee's performance). The exam is in two parts. First part is a presentation and discussion of selected parts of the course work. Second part is based on a drawn question from the course syllabus. Internal assessment.
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 Apply different filters (high-pass, low-pass, band-pass, notch) to remove unwanted components of digital signals Use the Fast Fourier Transform to analyse 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 analyse and interpret engineering problems Develop signal processing algorithms	At the end of the semester, the students will hand-in an assignment and the final exam will be based on this assignment. 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.
IT-EOS1	Embedded Operating	5	Having completed this course, students should be able to Account	Having completed this course, students should be able to Use basic	Having completed this course, students should be able to Implement	Oral Examination Individual oral examination based

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
	Systems		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 BeagleBone system. 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.	Linux commands and utilities. Select, install, configure and use tools needed for developing embedded systems. Execute a firmware upgrade on a BeagleBone system. Install and configure "off the shelf" software in Linux. 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.	shell scripts in BASH Design and implement IoT-devices, based on a 32-bit MCU platform with Linux	upon a subject found by draw. No preparation. Allowed tools: Laptop Internal assessment.
IT-ERP1	ERP systems SAP ABAP/4 Programming	5			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.	The exam is oral and it takes 20 minutes per student. The exam is in two parts. First part is a presentation and discussion of selected parts of the course work. Second part is drawn question from the theory of the course. Internal assessment.
IT-GMD1	Game Development (Horsens and Viborg)	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	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	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 devel-	Individual, oral examination, 15 minutes. The student will explain concepts and theories from the course, using their course project as reference The student will start with a prepared presentation of their project (5 minutes)

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			cross-platform game engine 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	applications on various hardware 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	oper position within a multidisciplinary game development pipeline, identifying and 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.	Allowed tools: All Internal assessment
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.	Oral Examination Individual oral examination (20 min. in total) based upon a subject found by draw. No preparation. Allowed tools: Laptop Internal assessment.
IT-IDX1	Interaction Design	5			Gain skills within interaction design and usability evaluation. You will achieve: knowledge of and experience in	The exam is oral and it takes 20 minutes per student. The exam is in two parts. First part is a presentation and discussion of selected parts of the course work. Second part is drawn question from the

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
					User eXperience Design (UX)including knowledge of and experience in participatory design workshops knowledge on planning, preparation, implementation, analysis, and documentation of user-based usability evaluation understanding of and practical experience with the interplay between usability evaluation and interaction design in an iterative design process	theory of the course. Internal assessment.
IT-LWA1	Internet of Things WAN's	5	After having completed this course, the students should have knowledge about: IoT protocols LoRaWAN network elements LoRaWAN technology LoRa device classes Sensors/Actuators OTA (OverTheAir) activation Personal activation Radio Propagation/Antenna Security IoT Testing	After having completed this course, the student should be able to: Analyse and explain IoT LoRaWAN problems and their context Select and argue for choice of methods and reflect critically on said methods Find and assess relevant literature within the problem domain Plan and present the result for an audience of engineers	After having completed this course, the student should be able to: Assess and delimit an IoT project Plan and structure the project within the set time limit Predict the preliminary steps in a systems 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	Oral examination based on one mandatory assignment. Internal assessment.
IT-MAL1	Introduction to Machine Learning	5	After having successfully completed the course, the student will have gained knowledge about theories, methods, techniques, tools, and applications within the following fundamental machine learning methods: predictive methods, e.g. regression and classification descriptive methods, e.g. clustering and factor analysis deep learning methods, e.g. neural networks. The students must be able to relate critically and reflectively to the above topics; In particular, it is important that they become proficient	After having successfully completed the course, the students should be able to apply the theories, methods and models from the above-mentioned areas to identify, analyse, evaluate and make suggestions for solving specific data-based issues. They must be able to argue for the relevance of the chosen theories, methods and models as well as for the proposed solution method. In addition, they must be able to reflect on the importance of the context in which the solution is included. Specifi-	After completion of the course, the goal is that the students have acquired the competences to: Make informed choices about the use of machine learning techniques Parametrisise machine learning algorithms for a given data material Design and develop a complete solution for a complex, realistic problem Communicate and discuss the solutions with professionals and non-specialists.	The course is evaluated based on two oral examinations. The first examination is a group exam in which the students make a 10 minute presentation about their group assignment. This is followed by approx. 20 minutes of discussion between the students and the examiners. This discussion will evolve around two of the main topics of the course. The group examination takes a total of 30 minutes. After the group exam, each student is then called for an individual 15 minute oral exam. This exam is

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			in selecting the right type of machine learning method for use in a given context.	cally, it is expected that after completion of the course the students will be able to: Understand and apply a number of machine learning methods for knowledge detection in both un-structured and structured data examples Understand and compare the algorithms behind different data mining and machine learning methods Match and possibly combine methods for practical use in a reasonable context.		a discussion about the two main topics that were not covered in the group examination. The student in not allowed to make a presentation at the individual oral exam. The 15 minutes include grading and feedback. The student is given one grade based on both the group exam and the individual exam. Internal assessment.
IT-MIX1	Mixed Reality	5	The students will develop knowledge about the use of hardware and sensors in developing systems in different settings, using the Unity3D-engine. Furthermore, the students will have an understanding of the reality–virtuality continuum, including: Achieving knowledge about where systems can be placed in the reality–virtuality continuum. Understanding how existing systems can be augmented At the end of the course, the students are expected to know how virtual and physical technologies can impact systems, and how virtual and real systems can be combined.	After the course, the students will have developed skills in Creating systems placed in physical or virtual realities, or in the space between them - the Mixed Reality space Using hardware and software to augment systems	At the end of the course, the students are able to implement Mixed Reality systems, and reflect on their placement in the reality–virtuality continuum.	Individual, oral exam, 20 minutes. At the exam, the student will present their course assignment in dialogue with the examiner. Internal assessment.
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 schemas and constraints in non-relational databases compare relational and different	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 adminis-	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 run-time environment to use for the data model.	Oral examination based on a question from the course syllabus based on the course assignment. Approximately 20 minutes (including discussion of examinee's performance) without preparation. Internal assessment.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
			non-relational approaches to database design.	trative tools to set up data replication administrative tools to set up sharding.		
IT-PCL1	Programming Concepts and Languages	5	Having 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 using F# and Python programming languages.			Three-hour written examination with marks according to the 7-point grading scale. Internal assessment.
IT-PME1	Process Management for ICT Engineering	5	After successfully completing the course, the students will have gained knowledge about: How to ensure quality in projects How to improve your project performance How to handle change management in a project.	After successfully completing the course, the student will be able to: Apply techniques and results from Capability Maturity Model Integration (CMMI) to solve challenges in project processes Apply techniques and results from Lewin model to handle change management in project Apply "How to break software" to prevent making mistakes in your project Be able to describe and make use of testing concepts Use of terminology to kick-start Bachelor project.	To complete this course the students must make hand-in: ”Test plan” document for a project ”Test Specification” document for a project ”Project relations to CMMI model” document for a project.	The exam is oral and it takes 20 minutes per student. The exam is in two parts. First part is a presentation and discussion of selected parts of the course work. Second part is drawn question from the theory of the course. Internal assessment.
IT-RTP1	Real-Time Programming, Interfacing and Electronics	5	Having completed this course, the student has gained knowledge in the below areas. Specifically, the student is able to: Understand the basic concepts of real-time programming Explain issues like deadlocks, priority inversion etc.	Having completed this course, the student should be able to: Write functioning real-time programs in C using FreeRTOS Analyse a simple real-time design for schedulability, deadlocks, utilization etc.	Having completed this course, students should be able to: 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-	Oral examination based on mandatory assignment. Internal assessment.

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
					time systems using FreeRTOS and C-programming understand low-level protocols, CRC etc.	
IT-SEP6	Semester Project 6 - Software Eng.	10	After successfully completing the course, the student will have gained knowledge about What the profession of Software Engineering is What knowledge and science are What role knowledge plays in engineering and computer science Essential theoretical problems and schools within philosophy of science What constitutes science, pseudo-science and non-science The concept of paradigms and paradigm shifts A basic understanding of ethics and ethical thinking within the scope of science GDPR What types of data can and should be collected from users	After successfully completing the course, the student will be able to Relate critically to empirical-analytical theory and among other things be able to discuss what knowledge is, how it is generated and how it relates to practice Reflect upon and enter into discussions about computer science perspectives in academic contexts Assess the relationship between scientific knowledge and practical experience in creating new technologies Describe types of knowledge and competences composing engineering practice Collect data from a user Discuss ethical dilemmas in a scientific context	After successfully completing the course, the student will have acquired competences in Reflecting upon their own role as knowledge creators in an information society Use their knowledge in a practical context, e.g. setting up an IoT infrastructure	Type of Examination: Students are assessed based on 1) One multiple choice test 2) One group project 3) One group examination The final grade is given according to above stated elements. The examination departs from the group project. Each group has approximately 10 minutes to present their project. During this time the examiners will not interrupt unless they have clarifying questions, i.e. if something is unclear during the presentation. The scope of the following examination is calculated as approximately 8 minutes per student. For instance, a group consisting of four members should expect 10 minutes for presenting the project and then 32 minutes of examination followed by 8 minutes of voting and feedback - totalling 50 minutes. During the examination, the examiners will ask questions. The students will then raise their hand if they wish to answer the question. The examiners will then choose a student to answer the question. The examiners are responsible for making sure that all students receive an equal amount of examination time. It is, however, the responsibility of the student to indicate that they wish to answer a question. If a student consistently

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
						fails to raise their hand, the examiners will interpret this as the student does not know the answer, and it will count negatively towards the overall grade and assessment. It is important that the students respect when another student is speaking which also means that you should not interrupt each other. Only one student should speak at a time. If you would like to elaborate upon something one of your fellow students has said, please raise your hand to indicate this Even though the exam is a group examination, students within a group may very likely receive different grades. There is no percentage on the group report vs. the oral performance, but the grade will be given based on a holistic assessment of your overall performance where the multiple choice test accounts for approximately 25% of the grade. External assessment.
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	The final exam is a 3 hour written exam and takes place at Campus Horsens. All supplementary materials and aids are allowed, e.g. using a computer as a reference work. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. The exam must be completed in the Jupyter Notebook environment and the answers must be submitted in Wiseflow. Internal assessment. The re-exam may be held as an

Code	Title	ECTS-points	Knowledge	Skills	Competencies	Assessment
						oral examination.
IT-SWA1	Single-Page Web Applications	5	The student should be able to define dynamic and static languages identify pitfalls of asynchronous programming explain the JavaScript object model including prototype model define functional programming describe the methods for client/server programming describe web sockets explain reactive programming	The student should be able to apply use object-oriented programming in JavaScript use functional programming in JavaScript apply common design patterns to document object models use callbacks and other higher-order functions for general programming in JavaScript apply asynchronous programming using Promises to implement client/server programming apply asynchronous programming using XmlHttp Request to implement client/server programming use web sockets to implement client/server communication use reactive programming including operators to implement client/server communication	At the end of the course, the students should be able to compare methods of object creation compare methods of inheritance relate functional programming to imperative programming in JavaScript compare patterns of GUI programming compare use of request/response, web sockets and reactive programming design and implement single-page application using JavaScript, web services, asynchronous programming, and client/server communication patterns	Oral examination based on a question from the course syllabus based on the course assignment. Approximately 20 minutes (including discussion of examinee's performance) without preparation. Internal assessment.
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 team work 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 ICT 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 behaviour analysis. Work successfully in a project group with the objective of solving a well-defined engineering problem.	Type of Examination: Oral examination with external assessment The basis of the evaluation is the reports, the solution of the ICT Engineering problem, and the oral examination. The student's ability to express oneself (in writing and orally) and to spell is part of the evaluation. Group presentation of the project (20 minutes). Individual examination of each member of the group (20 minutes). The individual examination typically starts from topics in the report and may involve all the topics from 1st to 7th semester.