Unfold your talent
VIA University College

Date:	June 2017
Rev.	May 2020
Rev.	August 2021
Rev.	November 2021
Rev.	August 2022
Rev.	February 2023 (electives)
Rev.	August 2023 (clarification exam prerequisites, 6th and 7th semester Change of prerequisite for obtaining the specialisation designation: Internet of Things)

Curriculum
Programme section

Bachelor of Engineering in Software Technology

Applicable for students enrolled August 2017 to February 2021.

NB! This curriculum is enforced from the autumn admittance of 2017.

Content

Introduction

In accordance with the Executive Order of the Bachelor Programme, the purpose is to equip the students with the following competences:

- Translate 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 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, both nationally and internationally, with people who have different educational, linguistic, and cultural backgrounds

In addition, the programme will qualify students to participate in further studies.

At VIA Engineering, we are problem and project oriented and focused on real world challenges. Keeping this focus will result in graduates with an education in the newest technologies that can be applied from the first day of employment, both in professional and research environments.

Programmes taught in English and an international environment is a characteristic of our engineering programmes. This profile creates a unique opportunity to train students to work nationally in an increasingly global market. Our lecturers have a broad practical experience, and they understand how to apply theory through courses, company visits, and projects for and in collaboration with companies.

To ensure the quality of the programme, the principles of the CDIO concept are applied ensuring that the individual courses are continuously reviewed, evaluated, and developed.

1 Identity of the Programme

VIA Software Technology Engineering 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 objectives of the programme are achieved primarily through:

- Interaction between theory and practice with focus on application oriented and practical engineering work. With an emphasis on project work the focus is on developing the student's professional, methodological, communicative, and personal skills
- Collaboration with businesses and research environments in connection with the development and educational activities
- Providing an international study environment, with all courses offered and taught in English for
- Danish and international students, with the possibility to carry out parts of the programme abroad
- The student's internship provides a means of exchanging knowledge and experience between the university college and the industry

2 Structure of the Programme

The programme is a full-time higher education programme.

The planned duration of the programme is 3 ½ years divided into 7 semesters and a total of 210 ECTS points. A single ECTS point represents 27.5 hours of study activity. Each year the student will be able to complete study activities corresponding to 60 ECTS points. Study activities are:

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

All compulsory and elective courses are either 5 or 10 ECTS point courses. Projects range from 5 to 15 ECTS points. The overall content, learning goals, evaluation methods, and tests of each course/project are described in this curriculum. More detailed descriptions can be found online.

Depending on the choice of electives, students will specialise in one of four areas:

- Internet-of-things
- Interactive Media
- AR/VR (augmented and virtual reality)
- Data Engineering

The programme is structured as illustrated below:

Semester/ Theme	Course	Course	Course	Course/ Project	Project	
7th semester Electives/ Specialisation	Elective course	Elective course	Elective course	**BPR2** Bachelor Project		
6th semester Electives/ Specialisation	Elective course	Elective course	Elective course	**BPR1** Bachelor Project Preparation	**SEP6** Semester Project	
5th semester Internship	**INP1** Engineering Internship					
4th semester Internet-of-Things	**AND1** Android Development	**ESW1** Embedded Software	**DAI1** Data Analytics Infrastructure	**ADS1** Algorithms and Data Structures	**INO1** Cross Disciplinary Innovation	**SEP4** Semester Project
3rd semester Heterogeneous Systems	**SDJ3** Software Development with UML and Java 3	**CAO1** Computer Architecture and Organisation	**DNP1** .NET Programming	**NES1** Networking and Security	**SEP3** Semester Project	
2nd semester Client/Server Systems	**SWE1** Software Engineering	**SDJ2** Software Development with UML and Java 2		**DBS1** Database Systems	**SEP2** Semester Project	
1st semester Single User Systems	**MSE1** Math for Software Engineers	**SDJ1** Software Development with UML and Java		**RWD1** Responsive Web Design	**SSE1** Study skills for Engineering Students	**SEP1** Semester Project

After the third semester, students can choose to transfer to the specialised AR/VR path in Viborg and complete their studies at the Interactive Design Centre (IDC), Viborg. If they do so, they will follow a specially adapted course plan illustrated in the following table.

Semester/ Theme	Course	Course	Course	Course/ Project	Project	
7th semester Electives/ Specialisation	**AVR3** Motion Capture and special device AVR	Elective course	Elective course	**BPR2** Bachelor Project		
6th semester Electives/ Specialisation	**AVR2** Advanced Augmented and Virtual Reality Technologies		Elective course	**BPR1** Bachelor Project Preparation	**SEP6** Semester Project	
5th semester Internship	**INP1** Engineering Internship					
4th semester AR/VR 101	**AVR1** Basic Technologies for AR/VR		**GMD1** Game Development	**DAP1** Digital Animation Production	**SEP4** Semester Project	
3rd semester Heterogeneous Systems	**SDJ3** Software Development with UML and Java 3	**CAO1** Computer Architecture and Organisation	**DNP1** .NET Programming	**NES1** Networking and Security	**SEP3** Semester Project	
2nd semester Client/Server Systems	**SWE1** Software Engineering	**SDJ2** Software Development with UML and Java 2			**SEP2** Semester Project	
1st semester Single User Systems	**MSE1** Math for Software Engineers	**SDJ1** Software Development with UML and Java			**SSE1** Study skills for Engineering Students	**SEP1** Semester Project

3 Content of the Programme

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

3.1 1st semester: Single User Systems

Topics

- Software Development with UML and Java (SDJ1)
- Responsive Web Design (RWD1)
- Math for Software Engineers (MSE1)
- Study Skills for Engineering Students (SSE1)
- Semester Project (SEP1)

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 2.

Course purpose:

Software Development with UML and Java (SDJ1)
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.
Responsive Web Design (RWD1)
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 skills for designing and developing responsive web sites for both PCs and mobile devices using basic web programming.
Math for Software Engineers (MSE1)
The main purpose of the course is to give the students the mathematical prerequisites to work with technical IT and specifically software engineering. With regard to the competency profile of the Software Engineering Programme, the focus of the course will be to • supply competences in analysing and generalizing algorithms and problems that occur in the con-text of software development • supply skills in expressing ones knowledge clearly and concisely • formalize statements in a logically and computationally correct manner • supply analytical problem-solving skills After having successfully completed the course, the students will have acquired a solid understanding of the mathematics used in software engineering, a clear analytical mindset, as well as skills in the methodology of software engineering. More specifically, the course covers the following topics: • Number Theory: o Number systems, including binary, hexadecimal, and decimal o Modular arithmetic o Prime numbers and factorization o Group Theory • Algebra: o Boolean algebra o Groups o Logic gates o Matrix algebra • Set Theory: o Sets o Functions • Combinatorics and probability theory
Study Skills for Engineering Students (SSE1)
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.
Semester Project (SEP1)
The purpose is to develop and document a single user system as well as demonstrate an acquisition of process skills.

Volume
30 ECTS credits

Learning objectives
Based on a case description from a real world scenario, students work in a team to analyse, design, implement, and document a small single user software system.

Knowledge
- Basic Java and JavaScript keywords and programming constructs
- Java types
- Java classes and objects
- Basic UML elements such as classes, fields, inheritance and other class relationships
- Web page design and layout
- Propositional and predicate logic
- Binary and hexadecimal numbering systems
- Boolean and relational algebra, sets, functions, and probability theory
- Project report structure and documentation styles
- Problem based learning, student responsibilities, team work and intercultural differences

Skills
- Translate formal or informal models into a programming language
- Use a modern IDE to implement, compile, and deploy small Java programs
- Use Bootstrap to add responsiveness to web pages
- Solve propositional and predicate logic expressions
- Translate between decimal, binary and hexadecimal numbering systems
- Describe program flow using activity diagrams
- Map mathematical types to Java data types

Competencies
- Use appropriate mathematical concepts and models to describe the solution to a computational problem
- Use appropriate UML diagrams such as use cases, use case diagrams, activity diagrams and class diagrams to analyse a problem and outline the design of a potential solution
- Use the Java programming language to implement a solution coherent with the analysis and design and/or mathematical model
- Use HTML, CSS, JavaScript, and the Bootstrap framework to design a webpage
- Document software development and process in a coherent manner

Exams
Number of tests and test forms. For detailed information and requirements see appendix **2.**

Courses	**ECTS**	**Evaluation**	**Examination** (re-examination may be oral)
Math for Software Engineers (MSE1)	5	Attendance (≥ 75%)	Written (4 hours) Internal examiner
Software Development with UML and Java (SDJ1)	10	Attendance (≥ 75%) Mandatory course activities completed	Oral (30 minutes) External examiner Final grade: 20% midterm, 80% exam
Responsive Web Design (RWD1)	5	Attendance (≥ 75%) 3 approved assignments	Written (2 hours) Internal examiner
Study Skills for Engineering Students (SSE1)	5	Attendance (≥ 80%) 3 approved tests	Passed/not passed

Semester Project (SEP1)	5	Project report handed in before deadline	Oral Group examination Internal examiner

3.2 2nd semester: Client/Server Systems

Topics

- Software Development with UML and Java 2 (SDJ2)
- Software Engineering (SWE1)
- Database Systems (DBS1)
- Semester Project (SEP2)

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 2.

Course purpose:

Software Development with UML and Java 2 (SDJ2)
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. The course will provide students with the qualifications needed to understand how to: • Implement solutions in Java using design patterns • Implement solutions in Java using threads • Develop client/server systems • Test software using various testing techniques
Software Engineering (SWE1)
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.
Database Systems (DBS1)
The main purpose of the course is two-fold. 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).
Semester Project (SEP2)
Completed the 1st semester Software Engineering course “Software Development with UML and Java 1” or a similar course.

Learning objectives

Analyse, design, implement, and test administrative systems that support access to shared data from multiple concurrent clients. Data is stored in a persistent and scalable manner.

Knowledge

- Design patterns
- Interfaces
- Threads and concurrency
- Monitors and semaphores
- Relational databases
- E/R modelling

- SQL
- RMI
- Unit testing
- The synchronized keyword in Java Version control using git

Skills

- Install and populate relational databases
- Implement design patterns in Java
- Use Java Threads to implement concurrency
- Implement the monitor concept in Java
- Use JDBC to store and retrieve data from a relational database
- Use RMI to implement concurrent access to shared data
- Implement Unit test for Java programs

Competencies

- Identify and implement abstract data types
- Recognize situations that calls for the use of relevant design patterns
- Design client/server architectures
- Support concurrent access to shared data
- Apply appropriate sorting algorithms to support scalability
- Identify and implement abstract data types
- Store object-oriented data in relational databases
- Map object-oriented concepts to entity/relationship concepts

Exams

Number of tests and test forms. For detailed information and requirements see appendix **2.**

Courses	**ECTS**	**Evaluation**	**Examination** (re-examination may be oral)
Software Engineering (SWE1)	5	Attendance (≥ 75%) Mandatory course activities completed	Oral Internal examiner
Software Development with UML and Java 2 (SDJ2)	10	Attendance (≥ 75%) Mandatory course activities completed	Oral External examiner
Database Systems (DBS1)	5	Attendance (≥ 75%) Mandatory course activities completed	Written (4 hours) External examiner
Semester Project (SEP2)	10	Project report handed in before deadline	Oral Group presentation Individual examination Internal examiner

3.3 3rd semester: Heterogeneous Systems

Topics

- Software Development with UML and Java 3 (SDJ3)
- Computer Architecture and Organization (CAO1)
- .NET programming (DNP1)
- Networking and Security (NES1)
- Semester Project (SEP3)

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 2.

Course purpose:

Software Development with UML and Java 3 (SDJ3)
The students should be introduced to basic theory of distributed systems and be able to design and implement a distributed system.
Computer Architecture and Organization (CAO1)
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.
.NET programming (DNP1)
The purpose is to qualify the student to describe and implement the basic concepts of the C# programming language and the. NET developer platform.
Networking and Security (NES1)
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.
Semester Project (SEP3)
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.

Learning objectives

Analyse, design, implement, and test heterogeneous software solutions including the use of the programming languages Assembly, Java, and C#.

Knowledge

- Web services
- Assembly language
- C# and. NET language and environment
- Computer architecture: Registers, program counter, ALU, conditional vs. unconditional branches 2's complement
- Test driven development (TDD)
- TCP/IP, DNS, NAT, routing
- Sockets
- Data encryption algorithms
- Authentication

Skills

- Build the ALU using gates
- Build (simulated) hardware to implement conditional and unconditional jumps
- Apply TDD development
- Use a modern IDE for constructing C# programs
- Implement the function concept using assembly language
- Write small assembly programs to solve simple hardware near computational problems
- Implement data encryption using Java
- Implement exchange of data using sockets in C# and Java

Competencies

- Asses which language is best to apply for a given problem
- Understand when to use a cross compiler and the basics of its operations
- Design software solutions taking into account the hardware on which to execute the software

- Select appropriate libraries & methods for authentication and encryption

Exams
Number of tests and test forms. For detailed information and requirements see appendix **2.**

Courses	**ECTS**	**Evaluation**	**Examination** (re-examination may be oral)
Computer Architecture and Organization (CAO1)	5	Attendance (≥ 75%) 2 approved assignments	Written (2 hours) Internal examiner
Software Development with UML and Java 3 (SDJ3)	5	Attendance (≥ 75%) Mandatory course activities completed	Oral Internal examiner
Networking and Security (NES1)	5	Attendance (≥ 75%)	Written (3 hours) Internal examiner
.NET programming (DNP1)	5	Attendance (≥ 75%)	Written (3 hours) Internal examiner
Semester Project (SEP3)	10	Project report handed in before deadline	Oral Group presentation Individual examination Internal examiner

3.4 4th semester

3.4.1 4th semester: Internet of Things (Horsens)

Topics
- Android Development (AND1)
- Embedded Software (ESW1)
- Data Analytics Infrastructure (DAI1)
- Algorithms and Data Structures (ADS1)
- Cross Disciplinary Innovation (INO1)
- Semester Project (SEP4)

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 2.

Course purpose:

Android Development (AND1)
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. Through weekly exercises, the course covers various topics, which will provide the students with the qualifications needed to: • Understand general mobile development principles • Utilize the Android SDK to develop mobile and ubiquitous computing applications • Define responsive and interactive user interfaces using the Android framework • Establish communication to external services and devices, as well as communicating in and between applications using the Android framework • Handle resources and persist data using the Android framework • Identify and implement various application architectures • Implement external services for real time database functionality, location-based services, and more • Test and publish applications

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.
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.
Algorithms and Data Structures (ADS1)
The purpose of the course is to qualify the student to: Design, implement and analyse different algorithms Become acquainted with different advanced data structures
Cross Disciplinary Innovation (INO1)
Innovation is integral to business success in the 21st century and in this course, students will explore the innovator's mind-set and apply innovation processes to solve real-world problems. Students will be introduced to creativity, creative thinking, innovation theory and methods, and the primary learning experience will be hands-on going through the different phases of the innovation process. Innovation is not only getting a good idea, but also actually turning that idea into products or services that can be sold and make a profit in a highly competitive global market.
Semester Project (SEP4)
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.

Learning objectives

Analyse, design, implement, and test a software solution including hardware sensors, an android-based user interface and a persistent multi-user backend infrastructure. The solution must contain self-constructed electronics, and make use of the Java, C#, C, and assembly programming languages.

Knowledge

- Innovation methods (e.g. double diamond)
- Real-time operating systems (RTOS)
- C-programming
- Cloud computing
- Android Development
- Build servers
- Regression testing
- Tree based data structures
- Recursive backtracking algorithms
- Graph algorithms
- Big O notation

Skills

- Implement complete Android applications using a modern Android development environment
- Setup and maintain a build server for a larger software project
- Setup and maintain automated regression testing
- Implement graph and backtracking algorithms in Java
- Implement RTOS-based applications in C

Competencies

- Generate alternative innovative ideas to engineering problems
- Design complete solutions comprised of both hardware and software
- Decide on appropriate quality assuring methods for a given software development project
- Recognize when to use backtracking or graph-based algorithms
- Prove and compare the big O time complexity of alternative algorithms
- Implement full-scale Internet of Things solution

Exams

Number of tests and test forms. For detailed information and requirements see appendix 2.

Courses	**ECTS**	**Evaluation**	**Examination** (re-examination may have another form at)
Android Development (AND1)	5	Mandatory course activities completed Attendance (≥ 75%)	Written In January exam 2022 the exam is oral External examiner
Embedded Software (ESW1)	5	Mandatory course activities completed Attendance (≥ 75%)	Written (3 hours) External examiner
Data Analytics Infrastructure (DAI1)	5	Course assignment handed in before deadline Attendance (≥ 75%)	Oral (20 minutes) Internal examiner
Algorithms and Data Structures (ADS1)	5	Attendance (≥ 75%)	Written (3 hours) Internal examiner
Cross Disciplinary Innovation (INO1)	5	Attendance (100%)	Written (30 min) Internal examiner
Semester Project (SEP4)	5	Project report handed in before deadline	Group exam with individual assessment. Group presentation (~30 minutes) followed by a joint examination of 3-4 students from the group at a time (~15 minutes per student).

3.4.2 4th semester: AR/VR (Viborg) up to and including spring 2020

Topics

- Introduction to Augmented and Virtual Reality (AVR1)
- Game Development (GMD1)
- Digital Animation Production (DAP1)
- Semester Project (SEP4V)

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 3.

Course purpose:

Introduction to Augmented and Virtual Reality (AVR1)
The purpose of the course is to provide students with knowledge, practical skills and competences to develop Augmented / Virtual reality experiences for various platforms. The course provides general knowledge about VR / AR regarding their respective histories, general concepts and general theory about current development techniques. The students will also be provided with knowledge and skills about hardware included in the curriculum.

Game Development (GMD1)

The purpose of this course is to provide the student with knowledge, skills and competences to develop games and interactive experiences using a real-time game engine.
The course provides a multidisciplinary perspective to game development, but is primarily rooted in the role of the software engineer, focusing on code architecture in a script-based environment.
Through weekly exercises and a project, the course covers various topics, which will provide the student with the qualifications needed to develop interactive experiences across various platforms including PC and mobile.

Digital Animation Production (DAP1)

This course is the students' first introduction to 3D, aiming to make the student comfortable with the software and its basic tools.

During the course, the student will get an understanding of the principles of working in 3D, an introduction in creating assets with texturing, lightning, rendering, animation and exporting data for 3D print.

The student will get a driver license for the makerspace, so he/she is able to do 3D printing and laser cutting.

Semester Project (SEP4V)

In this course the students will have the possibility to create their own projects. which means that there is a lot of freedom, but also responsibility.

The students will work in groups to create an AR/VR/XR application. They decide themselves which technology they will work with and to a certain extent what their application should contain. The course will last the whole semester and is divided into 3 parts:

1. Concept phase: This phase last around 3 to 4 weeks and will revolve around forming groups, trying hands-on with different AR/VR applications and creating 50+ different ideas for all types of different technologies which in the end should be reduced to a single great idea and the creation of a preliminary plan.

2. Development phase: In this phase the work on the project really begins, here the students will start creating their applications, this will involve self-learning but with guidance from the teacher that will help guide the project by setting goals, giving inputs and …

3. Project writing: In the last phase the focus will be to finish the applications and write the report about the project.

Previously projects have included the following; A simulation application, Games, An interactive experience and a tool for disabled people.

Volume

30 ECTS credits

Exams

Number of tests and test forms. For detailed information and requirements see appendix 3.

Courses	**ECTS**	**Evaluation**	**Examination** (re-examination may be oral)
Introduction to Augmented and Virtual Reality (AVR1)	10	Mandatory course activities completed	7 grade evaluation of the total number of assignments.
Game Development (GMD1)	5	Mandatory course activities completed	Oral (20 minutes) Internal examiner
Digital Animation Production (DAP1)	5	Course assignments handed in before deadline	7 grade evaluation of the total number of assignments.
Semester Project (SEP4V)	10	Project report handed in before deadline	Oral exam Group presentation Individual examination External examiner

4 5th semester: Internship

The internship (INP1) comprises a semester of 30 ECTS and is equivalent to 20 weeks of full time work. The internship period is either paid or unpaid and takes place either in a private or in a public company in Denmark or abroad.

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

In order to start the internship, the student must have passed a minimum of 110 ECTS. The student is responsible for finding an internship, which must be approved by VIA. A supervisor will be appointed to the student by VIA.

The student prepares a plan for the internship programme with a corresponding assignment formulation, in cooperation with the company. The basis for the assessment of internship is reports from the student to VIA and a presentation where the supervisor can ask detailed questions about the internship. The internship is assessed passed/not passed.

5 6th and 7th semester

5.1. 6th and 7th semester in Horsens

The 6th and 7th semester consist of a mandatory course (BPR1), elective courses, and projects. Students can select one of the three specialisations but 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 5.3.
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 5.1.3.
Internet of Things, up to and including spring semester 2023:	BEL1, HWP1, and at least one additional “Internet of Things” course from section 5.1.3.
Internet of Things, from autumn semester 2023:	At least 15 ECTS should be selected from the “Interactive Media” courses in section 5.1.3.
Data Engineering	BUI1 and at least 10 additional ECTS should be selected from the “Data Engineering” courses in section 5.1.3.

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.

5.1.1 6th semester

On the 6th semester, a compulsory semester project is made, which will support the learned theory from the elective courses by means of a practice-based project, consisting of the project phases of a product. Furthermore, the use of elements of advanced project methods is also included in the semester project.

Topics

- Bachelor Project Preparation (BPR1)
- Semester Project (SEP6)
- Electives

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 2.

Course purpose:

Bachelor Project Preparation (BPR1)
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. Each bachelor project group must consist of 2-3 students. One-person groups are not allowed! It is recommended to find the project in association with an external company or organization.
Semester Project (SEP6) (up to and including spring 2023)
Philosophy of Information Technology and Ethics: The continuous efforts made within information technology (IT) not only affect a diverse array of areas within the development of modern society but also affect them in many different ways. In much the same manner, IT itself is conditioned by its historical, technological, and social contexts. The purpose of this part of the course is to examine the (sometimes implicit) assumptions about knowledge and the creation of knowledge made by engineers and scientists within their field. The objective is to gain a deeper understanding of the way engineering and science is conducted, and a basic understanding of fundamentals business activities in an IT company. In addition, the course will enable the students to •account for the distinctive character of IT and its relation to other academic disciplines •account for what it means to be a Software Engineer and account for their own professional identity •discuss which scientific and ethical consequences the profession implies •reflect upon the profession's content, its history, its social and business-related function •Finally, the students will come to understand and master the knowledge-based challenges that one meets in a modern information society. Cloud Computing and DevOps: Cloud Computing is characterized by the fact that IT services are provided and used on the Internet. The use of Cloud Computing is growing very rapidly. Cloud Computing can provide businesses with greater flexibility, savings and new technical capabilities, but also sets new requirements for a variety of legal, personal data and security issues.

The purpose of this part of the course is to examine the concept of Cloud Computing and the strategic considerations for using Cloud Computing. Contents: The course comprises - among others - the following topics: •Philosophy of natural science •IT Ethics •Professional profile •Usability testing and collecting data •Cloud and cloud computing •DevOps •Linux
Semester Project (SEP6) (from autumn 2023)
The purpose of SEP6 is to develop and document a cross-organisational innovation project in collaboration with a company or institution.
Project (SPRAU) Primarily for international students admitted for a shorter period than one semester.
The purpose of the Project is to evolve the student's ability to solve a relevant Software Engineering problem and document the solution. The more specific problems chosen to be dealt with shall be connected to the student's background from her/his home university. In a group, students must be able to analyse, design, implement and test complex problems and be able to carry out well-documented and tested solutions.

Volume
30 ECTS credits

Exams
Number of tests and test forms. For detailed information and requirements see appendix **2.**

Courses	**ECTS**	**Exam prerequisites**	**Examination** (re-examination may be oral)
Semester Project (SEP6)	10	None	Type of exam: Oral group exam with individual assessment. Exam is based on project report, process report, source code and group video presentation, all of which must be handed in before deadline. Group presentation (~30 minutes) followed by a joint exam of 3-4 students from the group at a time (~15 minutes per student). Internal assessment Allowed tools: All. Re-exam: Students who fail a semester project must attend an information meeting at the end of the summer exam period. At this meeting, students will be notified about the process of the re-exam and students will form groups, if possible, in relation to the number of failed students. Based on the feedback the students have received after the ordinary exam, they must prepare a new project, or the failed project must be improved. Deadline for hand in of the project is mid-August (exact date will be informed at the meeting). There will be no guidance in the period up to hand in. Oral assessment of the project takes place before the start of the autumn semester.

Bachelor Project Preparation (BPR1)	5	Mandatory course activities completed	Type of exam: Exam in the form of a mandatory written assignment (project description) Internal assessment Re-exam: Assessment of a revised written assignment (project description)
Electives	15	Refer to section 5.1.3.	Refer to section 5.1.3.
Semester Project (SPRAU) (Modified version of SEP6) Primarily for international students admitted for a shorter period than one semester.	10	None	Type of exam: Group exam with individual assessment. Exam is based on the project report(s), which must be submitted before deadline and apply with the formalities criteria stated under the Software Engineering specific Guidelines. Group presentation approx. 5-10 minutes per student, followed by a joint exam with a joint discussion and individual question rounds for approx. 25 minutes per student including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the exam. Internal examiner.

5.1.2 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) where 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 conditions for starting the bachelor project (BPR2) are that the student has passed all courses in the 1st - 6th semester (courses totalling 180 ECTS, including the 30 ECTS internship). The bachelor project is prepared in groups of two or three persons.

Topics

- Bachelor Project (BPR2)

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 1.

Course purpose:

Bachelor Project (BPR2)
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. One-person groups are not allowed!

Volume

30 ECTS credits

Exams

Number of tests and test forms. For detailed information and requirements see appendix 2.

Courses	ECTS	Examination (re-examination may be oral)
Bachelor Project (BPR2)	15	Exam prerequisites: Passed all other courses of the bachelor programme. Type of exam: Oral group exam with individual assessment. Exam is based on the project report(s), uploaded in WISEflow according to deadline. Group presentation of the project (20 minutes) followed by a joint examination of 20 minutes times the number of group members, including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the examination. External assessment. Tools allowed: N/A Re-exam: 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. There is no supervisor attached when (re)doing the project.
Electives	15	Refer to section 5.1.3

5.1.3 Overview of elective courses

On the Software Technology Engineering Programme, the following elective courses are available at 6th and 7th semester for all students regardless of specialisation choice:

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

Electives (5 ECTS)	Course purpose	Assessment, valid from A23 (see also appendix 1) (All re-exams may be oral exam)
Programming Concepts and Languages (PCL1) Within the specialisation: None	The purpose of the course is to qualify the student to: - Understand various programming concepts, paradigms and get knowledge about how different paradigms appear in different programming languages - Get thorough knowledge about the functional programming paradigm - Apply different paradigms to specific problems in different languages	Exam prerequisites: None Type of exam: Individual written exam, 3 hours Internal assessment Tools allowed: All aids are allowed included access to online material. However, it is not allowed to use AI tools such as ChatGPT and similar AI and Machine Learning driven tools and chatbots. Re-exam: Same as the ordinary exam.
Applied Linear Algebra (ALI1) Within the specialisation: None	The purpose of the course is to equip the student with basic knowledge about linear algebra and its applications. This will enable the student to not only understand but also apply linear algebra in solving practical engineering problems. Skills in linear algebra are of high importance when dealing with scientific computing,	Type of exam: The final exam has two parts. • The first part is a Flowlock exam in WISEflow. • The second part is a WISEflow exam without Flowlock. The second part must be completed in the Jupyter Notebook environment and

	image processing graphics, robot technology, algorithmics, coding theory, and more. As an example, the founders of Google have cited their course in linear algebra as the backbone of Google's PageRank feature (i.e., ordering web pages after importance). The course familiarizes students with scalars, vectors, matrices, determinants, operations on vectors and matrices, and systems of linear equations in matrix form. The course also presents applications of matrix theory to linear models, including examples from engineering.	the answers must be submitted in WISEflow. The exam has a total duration of 4 hours. The student will not be able to access the second part before the first part is concluded. Each part has an equal weight in the final grade. Internal assessment Tools allowed: In the first part the students are allowed to use any notes, books, and/or other written/printed material and will have access to pdf files on their laptop. In the second part all supplementary materials and aids are allowed, e.g., using a computer as a reference work. It is not allowed, however, to use AI-tools such as CoPilot, ChatGPT, Bing, etc. as per the general VIA rules. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Re-exam: Re-exams may be oral.
Process Management for Software Engineering (PME1) Within the specialisation: None	The main purpose of the course is to provide students with the qualifications needed to understand the CMMI model and be able to transfer the CMMI level 1-2-3 knowledge into practical use in a project.	Exam prerequisites: None Type of exam: Ongoing assessment in the form of two group assignments of max two persons (assignment 1 and 2) and one individual assignment (assignment 3), weighing 25%, 30% and 45 % respectively. All assignment must have a grade of at least 02 and the final grade is a weighted grade according to the assignments individual weight. Internal assessment Tools allowed: N/A Re-exam: Individual oral re-exam, 20 minutes
Compiler Construction (CMC1) Within the specialisation: None	The students should be able to design a simple programming language, and design and implement a compiler for the language.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes. The exam is based upon a subject found by draw, and the answers are based on the project produced in the course and handed in according to deadline. Internal assessment. Tools allowed: All Re-exam: Same as the ordinary exam
Domain Centric Architecture (DCA1) Within the specialisation: None	The purpose is to qualify the student to understand and master the concepts and techniques of software architecture, test driven development, implementations of both. The course will provide students with the	Exam prerequisites: None Exam type: Individual oral exam, 20 minutes, without preparation.

	qualifications needed to understand how to: • Discuss various architectural styles • Implement several architectural patterns • Apply a subset of strategic and tactical Domain Driven Design patterns • Drive software development with automated tests	The exam is based upon course assignments, which must be submitted before the given deadline. The student will draw from a pool of known questions, and they will explain concepts and theories from the course, using the course work as reference. The student will start with a prepared presentation. Internal assessment. Tools allowed: N/A Re-exam: Same as the ordinary exam.
IT Security and Cryptography in Practice (SCP1) Within the specialisation: None	IT Security and Cryptography in Practice is an elective course at the Software Engineering Programme. It is offered in the spring and fall semesters. This course is student-driven and is focused on the real-world application of IT security in a practical environment. It includes experiencing how information and risk, threats and attacks, cyber security architecture and operations, secure systems hardening, and usability and cyber security management are applied to provide resilience in practical context. Students who do this course will obtain practical experience in the design, implementation, and evaluation of cyber security approaches.	Exam prerequisites: 3 mandatory assignments handed in: 1) A 1-page summary of their project idea. 2) A 1-page summary of their midterm seminar report. 3) A 1-page summary of their final report. If a student fails to meet one or more of the above mandatory assignments, the student will be given an extra assignment, to qualify for re-exam. The scope of this assignment depends on the scope of the missing requirements. Type of exam: The exam has ongoing assessment. Midway exam based on Midway Paper (30%) Final Exam based on Final Paper (70%) Internal assessment Tools allowed: All Re-exam: The re-exam consists of two parts: 1) A 1-page summary of each of the main topics in the course, incl. the student's own topic (10%) 2) A 20-minute oral exam based on Final Paper (90%) The student may choose to resubmit a revised version of the final paper. The main topics of the course are determined by the students at the beginning of the course and consists of the topics of their final paper.
Calculus (CAL1) Within the specialisation: None	In the course, the students attain knowledge about and practical experience in applying the methods and tools of calculus. Most importantly, the course will enable the student to apply differential and integral calculus in solving a wide range of problems.	Exam prerequisites: None Type of exam: Ongoing tests in the form of course assignments, (10%) and an oral exam (90%), 20 minutes, including everything. A selection of approximately 10 of the exercises from the course will form the basis for the exam. During the exam, the student will randomly draw and present one of these exercises. There is no preparation time. The exam will then evolve into a general discussion of the course curriculum.

		Tools allowed: The student is allowed to bring their notes to the oral exam, but these must be placed on the table during the exam. During the presentation, the student is allowed to consult their notes if they need to, but excessive use of the notes will count negatively towards the grade. During the discussion that follows the presentation, the student is not allowed to consult their notes. Internal assessment Re-exam: Same as the ordinary exam.
Single-page Web Applications (SWA1) Within the specialisation: Interactive Media	The purpose of this code is to learn to implement single-page web applications.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, without preparation. Exam is based on a question from the course syllabus and based on one or more of the course assignments. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam.
Interaction Design (IDX1) Within the specialisation: Interactive Media	To introduce students to newest principles of interaction design using UX and Usability	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, without preparation. Exam is based on a question from the course syllabus and based on the course assignment. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam.
Game Development (GMD1) Within the specialisation: Interactive Media	The purpose of this course is to provide the student with knowledge, skills and competences to develop games and interactive experiences using a real-time game engine. The course provides a multidisciplinary perspective to game development, but is primarily rooted in the role of the software engineer, focusing on code architecture in a script-based environment. Through weekly exercises and a project, the course covers various topics, which will provide the student with the qualifications needed to develop interactive experiences across various platforms including PC, mobile and interactive tables.	Exam prerequisites: None Type of exam: Written exam in the form of a course assignment, handed in before deadline. If the course assignment has been carried out as group work, it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each student must hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment. Internal assessment Tools allowed: N/A

		Re-exam: Same as the ordinary exam (individually or group). Students who fail the ordinary exam will be given a new deadline to hand in.
Digital Multi Media (DIM1) Within the specialisation: Interactive Media	To introduce students to basic principles of each media type - text, graphics, audio, animation and video - describing their digitization and progressing onto issues that arises when media are combined.	Exam prerequisites: None Type of exam: Written exam in the form of a course assignment, handed in before deadline. If the course assignment has been carried out as group work, it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each student must hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam (individually or group). Students who fail the ordinary exam will be given a new deadline to hand in.
XR Development (XRD1) Within the specialisation: Interactive Media	In this course students learn how to implement augmented- and virtual reality applications, reflect on their relevant use cases and gain an understanding of the underlying technology that enables the experiences.	Exam prerequisites: None Type of exam: Written assignment spanning the semester. Assessment is based on projects developed in groups where it must be clearly marked which parts of the applications each group member contributed to. Furthermore, each student must hand in an additional 1-2 pages (one page is considered 2400 characters) of individual reflections on the work they have done. Assessed by the lecturer and an internal assessor without oral presentation or defence. Tools allowed: N/A Re-exam: Same as the ordinary exam (individually or group assignment if possible). Students who fail the ordinary exam will be given a new deadline to hand in.
Basic Electronics (BEL1) Within the specialisation: Internet of Things	The student should acquire knowledge about basic electronic concepts and physical laws. Furthermore, the student should learn how to plan an experiment, perform it, analyse and evaluate the results and report all of this according to VIA ICT report-writing standard.	Exam prerequisites: None Type of exam: Ongoing assessment of three written assignments handed in according to deadline, weighing 50% Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation, weighing 50% Internal assessment

		Tools allowed: all Re-exam: Same as the ordinary exam.
Digital Signal Processing (DSP1) Within the specialisation: Internet of Things	The purpose of the course is to equip the student with basic knowledge about the fundamentals of Digital Signal Processing and its applications. Starting from the basic definition of a discrete-time signal, we will work our way through sampling, filter design, and Fourier analysis to build a basic DSP toolset. Signal processing is one of the fundamental theories and techniques to construct modern information systems. For example, audio, speech, and image processing, computer graphics, biomedicine all apply digital signal processing. In fact, digital signal processing is used to develop algorithms that can diagnose heart disease and can even be used to detect hostile drones. The course familiarizes the student with digital signals, sampling theory, digital filtering, the Fast Fourier Transform, power spectrum, and feature extraction.	Exam prerequisites: None Exam type: Individual oral exam, 20 minutes Exam is based upon an assignment handed in before deadline. The students will present the assignment in the form of a demonstration, followed by questions about the signal processing and feature extraction methods as well as the MATLAB programming. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam (new assignment).
Embedded Operating Systems (EOS1) Within the specialisation: Internet of Things	Students will acquire basic knowledge about the Linux Operating System and practical experience in development of an IoT-device based on a combination of self-made and Open-Source software.	Exam prerequisites: None. Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment Tools allowed: Laptop Course hardware kit Re-exam: Same as the ordinary exam.
Real-Time Programming (RTP1) Within the specialisation: Internet of Things	The main purpose of the course is to provide students with the qualifications needed to understand central concepts and characteristics about embedded real-time programming.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment Tools allowed: Laptop Re-exam: Same as the ordinary exam.
Internet-of Things WAN's (LWA1) Within the specialisation: Internet of Things (up to and including A22)	The purpose of this course is to make the students gain overall knowledge of a broad spectrum of IoT technologies with special focus on LoRaWAN and competing technologies. The students must make hand-on experience with state of the art LoRaWAN devices.	Oral exam (20 minutes) Internal assessment

Hardware Oriented Programming (HWP1) Within the specialisation: Internet of Things	The purpose of the course is: - To provide the student with knowledge about the technical details of an industrial microcontroller used for embedded systems from a programmer's point of view. - To qualify the student to implement simple low-level drivers for various hardware devices. - To qualify the student to implement low-level software for an embedded system in C. - To qualify the student to do Unit testing of embedded C.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment Tools allowed: Laptop Course hardware kit Re-exam: Same as the ordinary exam.
ERP Systems SAP ABAP/4 Programming (ERP1) Within the specialisation: Data Engineering	There are two main purposes of this course: The first purpose (36 lessons) is to introduce students to the aspects of analysis, design, coding and testing company specified programs in the SAP ABAP environment. The second purpose (12 lessons) is to give the students the opportunity to work further with a selected part of the SAP System: - Object Oriented programming in ABAP - Development of SAP S/4 HANA - SAP for production planning etc.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes. The exam is in two parts: First part is a presentation and discussion of selected parts of the course work (which consists of 2 mandatory written course assignments, handed in before deadline). Second part is drawn question from the theory of the course. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam. New assignments are accepted.
Introduction to Machine Learning (MAL1) Up to and including A23. Within the specialisation: Data Engineering	In the course the students attain knowledge and practical experience applying machine learning methods and tools to both structured and unstructured data problems. The students will gain in depth knowledge about tools and techniques, will be able to prepare data (through pre-processing) and use them to determine underlying structures as well as make predictions. The course evolves around the following four main topics: - Classification algorithms - Regression - Unsupervised learning - An introduction to Deep Learning and Neural Networks	Exam prerequisites: At the end of the course, the student must upload a 1-page summary of each of their 6 assignments as well as a 2-page summary of their group project. The summaries must include a brief description of: 1) the assignment problem 2) how the assignment was solved, e.g., data acquisition, data preparation, feature engineering, feature extraction, etc. 3) the algorithms that were used to solve the problem. 4) the performance of the final model 5) a reflection of the learning outcome of solving the assignment. Type of exam: The exam is a 20-minute oral examination that departs from one of the six assignments that the student made during the semester. The exam will also include an examination of the group project report. The final grade will be based on an overall assessment of the six assignments, the group project report, and the oral examination. Internal assessment Tools allowed: N/A

		Re-exam: Same as the ordinary exam.
Introduction to Machine Learning and AI (MAL1) From S24 Within the specialisation: Data Engineering	In this course, students will acquire both theoretical knowledge and practical skills in the application of machine learning methodologies to a spectrum of data types, encompassing both structured and unstructured datasets. The curriculum is designed to ensure that participants thoroughly understand and can adeptly utilize advanced tools and techniques essential for data preparation, preprocessing, and exploration. Students will be equipped to discern underlying structures and make informed predictions. Central to the course are four primary topics: - Classification: Understanding and categorizing data into predefined classes. - Regression: Predicting continuous outputs based on data input. - Clustering: Identifying the inherent groupings within datasets. - Dimensionality Reduction: Simplifying complex data structures without losing critical information.	Exam prerequisites: At the end of the course, the student must upload a 1-page summary of each of their 6 assignments as well as a 2-page summary of their group project. The summaries must include a brief description of: 1) the assignment problem 2) how the assignment was solved, e.g., data acquisition, data preparation, feature engineering, feature extraction, etc. 3) the algorithms that were used to solve the problem. 4) the performance of the final model 5) a reflection of the learning outcome of solving the assignment. Type of exam: The exam is a 20-minute oral examination that departs from one of the six assignments that the student made during the semester. The exam will also include an examination of the group project report. The final grade will be based on an overall assessment of the six assignments, the group project report, and the oral examination. Internal assessment. Tools allowed: N/A Re-exam: Same as the ordinary exam.
Machine Learning for Artificial Intelligence (MAL2) Within the specialisation: Data Engineering	This course explores the fundamental concepts, techniques, and applications of deep learning in the context of artificial intelligence (AI). This course is designed to provide students with a comprehensive understanding of how deep learning methods can be leveraged to solve complex AI problems.	Exam prerequisites: None. Exam type: Individual oral exam, 20 minutes without preparation. At the exam, the student will randomly draw one of the portfolio assignments. The exam will then take place as a discussion of this assignment, the students' group project and the curriculum in general. Internal assessment Tools allowed: The student is expected to bring their portfolio assignments and their final project to the oral exam, such that they are able to display and run their code. Re-exam: Same as the ordinary exam
Stochastic Modelling and Processes (SMP1) Up to and including S23	The ubiquitous presence of uncertainty and noise in the engineering sciences makes it mandatory to understand and quantify random phenomena. To achieve this goal the course will provide a solid introduction to the theory of stochastic processes. Special attention is given to	Exam prerequisites: None Type of exam: Individual written exam, 3 hours.

Within the specialisation: Data Engineering	applications and the student will model and analyse complex stochastic situations as encountered in practice. The applications include examples from various engineering fields such as information technologies and communications, signal processing, and more.	The exam must be completed in the Jupyter Notebook environment and the answers must be submitted in WISEflow. Internal assessment. Tools allowed: 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. Re-exam: Individual oral exam.
Stochastic Modelling and Processes (SMP1) From S24 Within the specialisation: Data Engineering	The ubiquitous presence of uncertainty and noise in the engineering sciences makes it mandatory to understand and quantify random phenomena. To achieve this goal the course will provide a solid introduction to the theory of stochastic processes. Special attention is given to applications and the student will model and analyse complex stochastic situations as encountered in practice. The applications include examples from various engineering fields such as information technologies and communications, signal processing, and more.	Exam prerequisites: None The final exam has two parts. • The first part is a Flowlock exam in WISEflow. • The second part is a WISEflow exam without Flowlock. The second part must be completed in the Jupyter Notebook environment and the answers must be submitted in WISEflow. The exam has a total duration of 4 hours. The student will not be able to access the second part before the first part is concluded. Each part has an equal weight in the final grade. Internal assessment Tools allowed: In the first part the students are allowed to use any notes, books, and/or other written/printed material and will have access to pdf files on their laptop. The student may bring their own calculator. In the second part all supplementary materials and aids are allowed, e.g., using a computer as a reference work. It is not allowed, however, to use AI-tools such as CoPilot, ChatGPT, Bing, etc. as per the general VIA rules. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Re-exam: Re-exams may be oral.
Business Intelligence (BUI1) Within the specialisation: Data Engineering	Business intelligence is the delivery of accurate, useful information to the appropriate decision makers within the necessary time frame to support effective decision making. The main purpose of the course is to equip the student to work with realistic business data using professional business intelligence tools in order to develop analytical solutions for businesses.	Exam prerequisites: None Type of exam: Oral exam based on the course assignment, which must be handed in before deadline. Approximately 20 minutes incl. discussion of examinee's performance, without preparation. Internal assessment. Tools allowed: N/A Re-exam: Same as the ordinary exam

No-SQL versus relational databases (NSQ1) Within the specialisation: Data Engineering	This course will provide students with knowledge of strengths and weaknesses of two fundamentally different approaches to database management systems.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes without preparation. Exam is based upon two course assignments handed in before deadline, and it is covering mandatory course work and theory covered in the course. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam

5.2 6th and 7th semester in Viborg up to and including 2021

5.2.1 6th semester

On the 6th semester, a compulsory semester project is made, which will support the learned theory from the elective courses by means of a practice-based project, consisting of the project phases of a product. Furthermore, the use of elements of advanced project methods is also included in the semester project.

Topics

- Advanced Augmented and Virtual Reality Technologies (AVR2)
- Bachelor Project Preparation (BPR1)
- Semester Project (SEP6V)
- Electives

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 3.

Course purpose:

Advanced Augmented and Virtual Reality Technologies (Viborg) (AVR2)
Qualify the students to create cross-platform AVR applications with focus on web and native AR. Provide knowledge of CAD and structured data file formats needed for exchange and storage of data in AVR applications. Qualify the students to setup and use a cloud-based system for communication and storage in AVR applications. Provide knowledge of render pipelines and shaders. Qualify the students to create their own shaders and edit existing ones.
Bachelor Project Preparation (BPR1)
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. It is recommended to find the project in association with an external company or organization.
Semester Project (SEP6V)
The purpose of this course is to introduce the students into a project which solution will require them to put newly obtained knowledge to practical use.

Volume
30 ECTS credits

Exams
Number of tests and test forms. For detailed information and requirements see appendix **3.**

Courses	**ECTS**	**Evaluation**	**Examination** (re-examination may be oral)
Advanced Augmented and Virtual Reality Technologies (AVR2)	10	Mandatory course activities completed	Individual oral exam with external co-examiner
Semester Project (SEP6)	10	Project report handed in before deadline	Oral exam External examiner
Bachelor Project Preparation (BPR1)	5	Mandatory course activities completed	Assessed on the basis of the project description and a draft of the analysis part of the project.
Electives	5	Refer to section 5.2.3	Refer to section 5.2.3

5.2.2 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) where 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 conditions for starting the bachelor project (BPR2) are that the student has passed all courses in the 1st - 6th semester (courses totalling 180 ECTS, including the 30 ECTS internship). The bachelor project is prepared in groups of two or three persons.

Topics
- Motion Capture and special device AVR (AVR3)
- Electives
- Bachelor Project Preparation (BPR2)

The learning objectives of the courses (knowledge, skills and competences) and test form are given in Appendix 3.

Course purpose:

Motion Capture and special device AVR (AVR3)
The purpose of this course is first to introduce the students to motion capture and how the data is used in game engines. Then the students are introduced to a chosen AVR device and must then code some advanced application parts using a relevant API and/or SDK.
Bachelor Project Preparation (BPR2)
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.

Volume
30 ECTS credits

Exams

Number of tests and test forms. For detailed information and requirements see appendix **3.**

Courses	**ECTS**	**Evaluation**	**Examination** (re-examination may be oral)
Augmented and Virtual reality technologies III (AVR3)	5	Mandatory course activities completed	Assessed on the basis of 3 written assignments. Internal assessment.
Bachelor Project (BPR2)	15	Project report handed in before deadline	Oral Group presentation Individual examination External examiner
Electives	10	Refer to section 5.2.3	Refer to section 5.2.3

5.2.3 Overview of elective courses

See also section 5.1.3

Advanced Unity Development (AUD1)
The purpose of this course is to enable the student to develop optimized games and interactive experiences. The course will provide knowledge and practical skills in using Unity's Data Oriented Technology Stack to develop games and experiences, along with additional tools for optimizing C# code.
Physical Computing (PCO1)
The purpose of this course is to introduce the students to developing interactive applications which interfaces to physical computing using Arduino and sensors. The students must also use 3D printer and laser cutter to get experience in using the products and try to cut in different materials to get a feeling for possibilities in using the technologies.
Advanced Game Development (GMD2)
The purpose of this course is to enable the student to develop games and interactive experiences in a structured and maintainable manner. The course is rooted in the software development aspect of game development, focusing on code architecture, design patterns, complexity, algorithms and data structures that are commonly used. Through weekly exercises and a project, the course covers various topics, which will provide the student with the qualifications needed to develop larger interactive experiences with a maintainable codebase.

Courses (5 ECTS)	**Specialisation**	**Evaluation**	**Examination** (re-examination may be oral)
Physical Computing (PCO1)	5	Mandatory course activities completed	Oral exam Internal examiner
Advanced Game Development (GMD2)	5	Mandatory course activities completed	Oral exam Internal examiner
Advanced Unity Development (AUD1) or elective offered in Horsens	5	Mandatory course activities completed	Oral exam Internal examiner

6 Title and issue of diploma

Graduates who have completed the studies in compliance with this curriculum and the joint regulations for VIA Engineering are awarded the title Bachelor of Engineering in Software Technology Engineering.

Upon completion of the programme, VIA University College issues a diploma indicating title, programme, specialisation if relevant, and information about the results of the grades obtained. Furthermore, the diploma contains information about the bachelor project. In addition, the admittance level on which the graduate was admitted to the programme is noted.

Should the education be discontinued, a transcript of records is issued.

Appendix 1: Courses for the Software Engineering Programme

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
IT-MSE1	Mathematics for Software Engineering	5	After having successfully completed the course, the student will be able to Describe fundamental concepts in set theory and Boolean algebra Outline the basic assumptions of group and number theory Read and use mathematical notation in the context of software development Summarize key aspects of elementary probability theory and counting techniques	The student will also be able to Read and construct mathematical arguments Design algorithms for solving simple problems Compare number systems and convert from one system to another Represent numbers in terms of modular arithmetic Use factorization algorithms to enhance computational performance Identify and describe mathematical functions as well as functions in programming	And having successfully completed the course the student will be able to Decompose complex problems into simple logical and mathematical components Perform problem analysis in a software development context.	The course is evaluated based on a 4 hour written final exam. The exam is handwritten. Except for a calculator, no electronic aids are allowed (e.g. laptops, phones, tablets, etc.). Apart from this, the students are allowed to use notes, books, and other written/printed material. Any type of communication between students or between a student and an external party will be considered a violation of the exam rules.
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. Implement exception handling for different types of exceptions Implement persistence in text and binary files Construct simple event-based GUI applications Construct Java source code documentations Interpret UML class diagrams, and construct corresponding Java code	The student should be able to: Exemplify and discuss basic object-oriented concepts, including encapsulation, relationships, inheritance and polymorphism Implement small scale systems from UML class diagrams, and construct related GUIs	Individual oral examination based upon a subject found by draw. No preparation. Oral examination where the student will pick an unfamiliar programming exercise at random. The student must explain the UML involved and demonstrate how to perform the programming task using a laptop and/or the blackboard. The time allotted for the examination is 30 minutes including assessment. The grade for the oral examination counts for 80 % of the final grade while the remaining 20 % comes from a test conducted in the middle of the course. Allowed tools: All External examiner.
IT-	Responsive Web	5	Having completed this course,	Having completed this course,	Having completed this course,	Written Examination

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
RWD1	Design		students will have the knowledge to: Describe the different file formats used in web development and their purpose. Reproduce webpage layouts using HTML5 and CSS3 when presented with images/screenshots of other websites. Select appropriate attributes for HTML5 elements. Explain the difference between responsive and non-responsive websites. Test HTML5 files for errors using the W3C markup validator. Account for the difference between the JavaScript and Java programming languages.	students will have the skills to: Create web sites using Hyper Text Markup Language (HTML5). Use simple and advanced CSS3 selectors and properties to style webpages. Apply the Bootstrap grid framework to create responsive websites. Utilize the Bootstrap classes to apply styling to responsive websites. Implement JavaScript functions to add functionality to websites. Use XMLHttp Request to read content from an external source and integrate this content into a website. Select HTML elements and apply jQuery animations to the selected elements to make websites interactive.	students will be able to: Design and implement platform independent web applications.	Duration: 2 hours Digital written examination (2 parts): Part 1: Multiple choice questions 30 minutes without aids Part 2: Short answer questions 90 minutes (explaining and writing code) with all aids, including internet connection Internal examiner
IT-SSE1	Study Skills for Engineering Students (ICT)	5	The student should be able to: Explain the study activity model, SOLO taxonomy, VIA Engineering's DNA Explain the strengths and weaknesses of Problem-Based Learning (PBL) Outline the stages of team development (such as the Tuckman stages) Identify motivational factors in a group setting Define culture's influence on values and norms and outline cultural traits that can influence team work in a project Define the project phases, including problem analysis, problem formulation and project planning Describe the role of the supervisor and project facilitation in general List the features of academic and technical writing and understand	The student should be able to: Apply good study techniques for planning, reading and note-taking in an intentional manner Prepare and deliver oral presentations using different rhetorical techniques Communicate correctly taking target audience into consideration Characterise typical group behaviour and group roles Apply an appropriate project methodology based on the VIA Software Engineering guidelines	The students should be able to: Reflect on active learning and how take responsibility for own learning Analyse and apply team dynamics such as communication, motivation, decision-making and conflict resolution Reflect on the importance of work style and behaviour, team roles and culture Generate a project outcome (report, appendix, etc.) that demonstrates effective communication skills	Approval/non-approval. Students who fail to comply with the above approval criteria, will fail the course (use one exam attempt) and be registered for a 20 minute oral re-exam (2nd exam attempt). The competences achieved in this course will be assessed at the project exams.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			the concept of plagiarism Define the characteristics of reliable sources (source criticism)			
IT-SEP1	Semester Project: Single User System	5	The student will use the knowledge acquired in SDJ1, SSE1 and RWD1.	The student will achieve the skills to: 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 Document system development and process using VIA Software Engineering's Project Guidelines and report templates Describe how to use your system in a user guide Plan and deliver a coherent oral presentation of the project Discuss the importance of work style and behaviour, team roles and culture	The student will be able to: Demonstrate the connection between the different stages in software development Apply, discuss and reflect on team dynamics such as communication, decision-making and motivation	The basis of the examination is the project report which should be done as a group project. First the entire group presents the project (5 minutes per person). Then a group examination will follow in accordance with the guidelines. It must be stated in the report which parts and subjects each member is responsible for. 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, 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-SWE1	Software Engineering	5	The student should be able to account for:	The student should exemplify:	The student should be able to:	IF you are full degree Software Engineering student:

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			Abstraction UML S.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	Analysis of a problem and document the analyse- and design-process in UML Practical use of UML Practical 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	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	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 The grade for SWE 1 and SEP 2 will be the same, but be given as two grades If a student fails the SWE1 or the SEP2 part of the exam, both courses are failed 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.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						ENDIF The course must be passed before graduation.
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 JavaTest 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	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 examiner. The course must be passed before graduation.
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 Modeling 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 implement simple mechanisms to deal with concurrency control	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	Oral Examination The examination is held as a 20 minutes examination all inclusive. The student will be examined in a randomly picked topic from the course syllabus as well as one or more course assignments. External Exam.
IT-SEP2	Semester Project: Client/Server	10	The student will be able to: Describe the relation between	The student will be able to:	The student will be able to:	Oral Examination

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
	System		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	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 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	Capture requirements, analyse, design, implement and test a client/server system using UML, Java and SQL Create and argue 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 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	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 SEP2The grade for SWE1 and SEP2 will be the same but be given as two grades. Allowed tools: Laptops, notes Internal examiner. Please note that re-examinations may take a different form than the ordinary exams. 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.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				form		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.
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.	Written examination. Duration: 2 hours. Allowed tools: Course literature according to the course description Personal notes Internal examiner. Examination is digital, and the students are required to bring a laptop that are tested and ready for use with WISEflow and FlowLock. The course must be passed before graduation.
IT-SDJ3	Software Development of Distributed Systems	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	Oral Examination The examination 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.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						Allowed tools: All Internal 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.	Written examination. Duration: 3 hours. Allowed tools: Course literature according to the course description Personal notes on paper Laptop (no access to local files or general internet) Internal examiner.
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. 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 Implement console applications, web applications and web services with Server-side C#-programming Data persistence using object-relational mapping User management, including authentication and authorization Create and consume class libraries Consume and expose RESTful web services Navigate and use the managed. NET API Reflect on security concerns of. NET applications Deploy. NET applications Use a command-line interface (CLI) toolchain Compare object-relational mapping to traditional data access	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	Written examination Duration: 3 hours Programming exercises; with all aids, including internet connection Internal examiner Please note that re-examination may take a different form than the ordinary exam

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				techniques Explain the difference between client-side and server-side programming		
IT-SEP3	Semester Project: Heterogeneous System	10		The student should be able to Formulate a project description for a 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	The student should be able to Analyse, design, implement and test a 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.	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. Allowed tools: All Internal exam.
IT-AND1	Android Development (up to and including S22)	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 SQLite Networking 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	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 The student must hand in an app developed throughout the course in order to attend the written exam. If the app is not accepted, the student cannot attend the written exam. Type of Examination: The written exam is a one-hour multiple choice test conducted in WISEflow. Re-exams: Any re-examination will be conducted orally based on an application developed by the student. The student is expected to present subjects from the curriculum by showing source code

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				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		from the application (~5 min). This is followed by questions from the examiners (~10 min), which may include live coding while explaining the underlying theory.
IT-AND1	Android Development (only A22)	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 SQLite Networking 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 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. The student must hand in an app developed throughout the course in order to attend the written exam. If the app is not accepted, the student cannot attend the written exam. Type of examination Written assignment spanning the semester. Evaluation is based on an application developed as a group project, where it must be clearly marked which parts of the application each group member contributed to. Furthermore, each group member must hand in an additional 1-2 pages (one page is considered 2400 characters) of individual reflections on the work they have done. External assessment. Re-examination Students who fail the ordinary examination must form groups and conduct a new project the following semester without any guidance allocated.
IT-ESW1	Embedded Software	5	The student should be able to account for:	The student should be able to exemplify:	The student should be able to:	Three hour written exam.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			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	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	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	Allowed tools: Laptop External examiner. The course must be passed before graduation.
IT-DAI1	Data Analytics Infrastructure (up to and including S22)	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 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.	Having completed this course, students should be able to: Design and implement data models for integrating multi-source data, including dimensional data modelling, for structured and semi structured data Design and implement data models for time-variant data Design, implement and test systems for data acquisition, validation, integration and delivery from multiple sources and platforms Design, implement and test basic descriptive statistical analysis on integrated data Design, implement and test basic visualizations and graphs of data and analysis results.	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.	Oral examination, covering mandatory course work and theory covered in the course. Duration (grading included) app. 20 min/ 5 ECTS. Permit criteria for attending examination Mandatory course activities completed Course assignment handed in before deadline Type of exam: Oral examination Individual oral examination without preparation based upon course assignment(s) Allowed Tools: All Internal Examiner
IT-DAI1	Data Analytics Infrastructure (only A22)	5	Having completed this course, students should be able to describe basic techniques within the field, and argue the choice and applicability of these for different use scenarios. This includes: • Application of analytical data	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	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 • The student must have an attendance of at least 75% in order

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			processing, and differences to transactional processing • Types of analytical data processing, such as reporting and visualization • Sources of data for analytical processing • Server and locally hosted platforms for data storage and analytical processing • Modelling techniques for designing data models for integration of multi-source data, including structured, semi-structured and unstructured data, and for modelling time-variant data/history • Design of systems for data acquisition, validating and cleansing data, integration and publishing of data.	models for time-variant data • Design, implement and test systems for data acquisition, validation, integration and delivery from multiple sources and platforms • Design, implement and test basic descriptive statistical analysis on integrated data • Design, implement and test basic visualizations and graphs of data and analysis results. • Give relevant peer feedback on handins and exercises throughout the semester	• Evaluate and act upon peer feedback	to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam. Type of exam Individual oral examination without preparation based upon course assignment(s), covering mandatory course work and theory covered in the course. Duration (grading included) app. 20 min/ 5 ECTS. Allowed Tools: All Internal Examiner
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 non-linear 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	Prerequisites for exam: 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. Written examination Duration: 3 hours Allowed tools: • Course literature according to the course description • Personal notes • Internal examiner Please note that re-examination may take a different form than the ordinary exam.
IT-INO1	Engineering Innovation Weeks (Software Engineering)	5	After having successfully completed the course, the students will have gained:	After having successfully completed the course, the students will be able to:	After having successfully completed the course, the students will have gained competences in:	Group presentation of project in both shared-weeks as well as mono-weeks.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			An understanding of innovation and its uses within the field of engineering Knowledge about Design Thinking (double diamond) process Knowledge about how to create a systematic and measurable progress in innovation tasks	Engage in innovative processes in a cross-/inter-/multidisciplinary setting Conceive, plan, and execute innovative ideas Work methodically with innovation Collect and apply relevant information about technologies, markets and end users	Introducing innovative ideas into project work Contributing own professional skills in teams with the objective of solving problems by using innovative processes and models Clarifying multidisciplinary group competencies	
IT-SEP4	Semester Project 4	5	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/ dimensional modelling Using Extract, Clean-up, Transform and Load data flows	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 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	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 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	Permit criteria for attending examination: Project- and process report handed in before deadline Type of examination: Group exam with individual assessment. Group presentation (~30 minutes) followed by a joint examination of 3-4 students from the group at a time (~15 minutes per student). Individual grades are given based on an overall assessment of the submitted work (50%) as well as the individual´s performance during the examination (50%). 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, or the failed project must be improved.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						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 (ICT)	30	The student must: gain knowledge of theory, methodology and practice within a profession or one or more fields of study be able to understand and reflect on theories, methodology and practice be aware of non-technical – societal, health and safety, environmental, economic and industrial – implications of engineering practice	The student must: be able to apply the methodologies and tools of one or more fields of study and to apply skills related to work within the field/fields of study or profession be able to assess theoretical and practical problems and to substantiate and select relevant solutions be able to communicate professional issues	The student must: be able to handle complex and development-oriented situations in study or work contexts be able to independently participate in professional and interdisciplinary collaboration with a professional approach be able to identify own learning needs and to organise own learning in different learning environments promote an engineering-oriented approach during the remaining semesters on the Bachelor programme develop personal skills required for the professional career as engineer form the basis for developing personal/professional network.	
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 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 exam: Students are assessed based on 1) One multiple choice test 2) One group project 3) Group and Individual Video Presentations Apart from the midway multiple-choice test, there is no exam in the course.
IT-	Semester Project	10	After having completed this course,	After having completed this course,	After having completed this course,	Type of exam:

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
SPRAU	(short course)		the students must master the knowledge about: • Searching and scoping relevant project information • Project and team work planning • Communication and documentation skills • Testing	the student must be able to: • Identify and justify problems and their context • Select and argue for choice of method and reflect critical on methods • Find and assess relevant literature within the problem domain • Present the result for an audience of engineers	the students must be able to: • Describe and delimit a large Software Engineering Project • Select and use relevant theories and methods to solve the problem • Plan and structure the project within the project time frame • Initiate the preliminary steps in a system development process, leading to a clearly defined requirements capture, use cases as well as object and behaviour analysis. • Work successfully in a project group with the objective of solving a well-defined engineering problem.	Group exam with individual assessment. Exam is based on the project report(s), which must be submitted before deadline and apply with the formalities criteria stated under the Software Engineering specific Guidelines. Group presentation approx. 5-10 minutes per student, followed by a joint exam with a joint discussion and individual question rounds for approx. 25 minutes per student including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the exam. Internal examiner.
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	The basis of the evaluation is three mandatory assignments: Group Description and Group Contract Project Description Requirements/User Stories Type of exam: None
IT-BPR1	Bachelor Project Preparation (from A22)	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 exam: Mandatory course activities completed Type of Exam: Assessed on the basis of one mandatory assignment, the Project Description. Internal assessment Re-exam: Re-submission of a revised Project Description
IT-BPR2	Bachelor Project 2	15	After having completed this course, the students must master the	After having completed this course, the student must master to: Identify	After having completed this course, the students must be able to:	External examination. The basis of the evaluation is the

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			knowledge about: Searching and scoping relevant project information Project and team work planning Communication and documentation skills Testing	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	Describe and delimit a large Software Engineering Project Select and use relevant theories and methods to solve the problem Plan and structure the project within the BPR2 time frame Initiate the preliminary steps in a system development process, leading to a clearly defined requirements capture, use cases as well as object and behaviour analysis. Work successfully in a project group with the objective of solving a well-defined engineering problem.	reports, the solution of the Software 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. Type of exam: Oral examination. 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.
IT-BPR2	Bachelor Project 2 (from A23)	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 Software Engineering Project Select and use relevant theories and methods to solve the problem Plan and structure the project within the BPR2 time frame Initiate the preliminary steps in a system development process, leading to a clearly defined requirements capture, use cases as well as object and behaviour analysis. Work successfully in a project group with the objective of solving a well-defined engineering problem.	Exam prerequisites: Passed all other courses of the bachelor programme. Type of exam: Oral group exam with individual assessment. Exam is based on the project report(s), uploaded in WISEflow according to deadline. Group presentation of the project (20 minutes) followed by a joint examination of 20 minutes times the number of group members, including voting. Individual grades are given on the basis of an overall assessment of the submitted work as well as the individual´s performance during the examination. External assessment. Tools allowed: N/A Re-exam: Based on the feedback the students have received after the ordinary

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						exam, they must prepare a new project, or the failed project must be improved. There is no supervisor attached when (re)doing the project.
IT-ALI1	Applied Linear Algebra	5	After successfully completing the course, the student will have gained knowledge about: What a vector space is, and How a linear representation of such spaces can be analysed using matrix operations Application of linear algebra in engineering	After successfully completing the course, the student will be able to: Apply techniques and results from linear algebra to solve problems in linear systems, matrices, vector space, orthogonality, eigen vectors, and eigenvalue Apply theory to analyse basic theoretic tasks within the below mentioned topics Express mathematically correct arguments Use mathematical terminology and symbol language	After successfully completing the course, the student will have acquired competences in: Applying linear algebra to the study of various phenomena in engineering science Using matrices to solve concrete problems Using vector operations to solve concrete problems Applying methods and results from linear algebra in the solution of engineering problems	Grading will be done according to the 7-scale, using an internal examiner. In order to qualify for the exam, the students must have the compulsory activity approved. Type of exam: 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. 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-ALI1	Applied Linear Algebra (from A23)	5	After successfully completing the course, the student will have gained knowledge about: What a vector space is, and How a linear representation of such spaces can be analysed using matrix operations Application of linear algebra in engineering	After successfully completing the course, the student will be able to: Apply techniques and results from linear algebra to solve problems in linear systems, matrices, vector space, orthogonality, eigen vectors, and eigenvalue Apply theory to analyse basic theoretic tasks within the below mentioned topics Express mathematically correct arguments Use mathematical terminology and symbol language	After successfully completing the course, the student will have acquired competences in: Applying linear algebra to the study of various phenomena in engineering science Using matrices to solve concrete problems Using vector operations to solve concrete problems Applying methods and results from linear algebra in the solution of engineering problems	Exam prerequisites: None Type of exam: The final exam has two parts. • The first part is a Flowlock exam in WISEflow. • The second part is a WISEflow exam without Flowlock. The second part must be completed in the Jupyter Notebook environment and the answers must be submitted in WISEflow. The exam has a total duration of 4 hours. The student will not be able to access the second part before the first part is concluded. Each

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						part has an equal weight in the final grade. Internal assessment Tools allowed: In the first part the students are allowed to use any notes, books, and/or other written/printed material and will have access to pdf files on their laptop. In the second part all supplementary materials and aids are allowed, e.g., using a computer as a reference work. It is not allowed, however, to use AI-tools such as CoPilot, ChatGPT, Bing, etc. as per the general VIA rules. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Re-exam: Re-exams may be oral.
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	Permit criteria's for attending examination: * Mandatory course activities completed. Duration (grading included) approx. 20 min/ 5 ECTS. Type of exam: Individual oral examination based upon a subject found by draw. No preparation Duration 0. 3 hours Allowed tools: All Internal examiner
IT-BEL1	Basic Electronics (from A23)	5	Having completed this course, students should have understanding of: Statistics,	Having completed this course, students should be able to: Simulate analogue electronic	Having completed this course, students should be able to: Design simple electronic circuits for	Exam prerequisites: None

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			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	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	measurement systems using amplifiers and filters Analyse experiment results, using statistical calculations and methods Write reports to document engineering experiments	Type of exam: Ongoing assessment of 3 written assignments handed in according to deadline, weighing 50% Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation, weighing 50% Internal assessment Tools allowed: all Re-exam: Same as ordinary exam.
IT-BUI1	Business Intelligence	5	Students will obtain knowledge about understanding, reading, and presenting data from a dimensional model (such as a star schema or data cube) and other data models. Knowledge about building data products for operational vs real-time systems	Data migration using data integration tools Create Data pipelines to cleanse data and move it into a data ware-house Create KPIs and measures Create data analyses, presentations and dashboards with Business Intelligence tools Create data structures for analysis purposes with selected tools Create, deploy and manage reports	Evaluate pros/cons of different BI products, architectures and approaches	Internal examination. In order for the student to qualify for the examination, the course assignment must have been handed in and approved. Type of exam: Oral examination based on the course assignment. Approximately 20 minutes incl. discussion of examinee's performance, without preparation.
IT-BUI1	Business Intelligence (from A23)	5	Students will obtain knowledge about understanding, reading, and presenting data from a dimensional model (such as a star schema or data cube) and other data models. Knowledge about building data products for operational vs real-time systems	Data migration using data integration tools Create Data pipelines to cleanse data and move it into a data ware-house Create KPIs and measures Create data analyses, presentations and dashboards with Business Intelligence tools Create data structures for analysis purposes with selected tools Create, deploy and manage reports	Evaluate pros/cons of different BI products, architectures and approaches	Exam prerequisites: None Type of exam: Oral exam based on the course assignment, which must be handed in before deadline. Approximately 20 minutes incl. discussion of examinee's performance, without preparation. Internal assessment. Tools allowed: N/A Re-exam: Same as the ordinary exam
IT-CAL1	Calculus	5	After having successfully completed the course, the student will have gained knowledge about the theory,	Upon completion of this course, students will be able to: - Define and interpret functions,	Upon completion of this course, the goal is that the students have acquired the competences to:	Exam prerequisites: The student must hand in all mandatory assignments to qualify

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			techniques and tools of calculus, in particular knowledge about: - Functions - Limits and continuity - Derivatives - Integrals - Infinite series and sequences - Partial derivatives - Multiple integrals - Differential equations	including calculating limits of functions and the concept of continuity - Calculate and interpret ordinary and partial derivates of real functions. - Calculate and interpret definite and indefinite integrals of real functions. - Perform calculations pertaining to infinite series and sequences. - Solve differential equations.	- Make informed choices about the use of differential and integral calculus. - Apply the tools of calculus to real-world problems. Communicate and discuss the theory, tools and techniques of calculus.	for the exam. Exam type: The course is evaluated based on an oral examination, which will take 20 minutes including everything. A selection of approximately 10 of the exercises of the course will form the basis for the exam. These exercises will be selected in the last teaching session. During the exam, the student will present one of these exercises, which will be chosen randomly. There is no preparation time. The exam will then evolve into a general discussion of the course curriculum Tools allowed: The student is allowed to bring their notes to the oral exam, but these must be placed on the table during the examination. During the presentation, the student is allowed to consult their notes if they need to, but excessive use of the notes will count negatively towards the grade. During the discussion that follows the presentation, the student is not allowed to consult their notes. Re-exam: Please note that re-examinations may take a different form than the ordinary exams.
IT-CAL1	Calculus (from A23)	5	After having successfully completed the course, the student will have gained knowledge about the theory, techniques and tools of calculus, in particular knowledge about: - Functions - Limits and continuity - Derivatives	Upon completion of this course, students will be able to: - Define and interpret functions, including calculating limits of functions and the concept of continuity - Calculate and interpret ordinary and partial derivates of real	Upon completion of this course, the goal is that the students have acquired the competences to: - Make informed choices about the use of differential and integral calculus. - Apply the tools of calculus to real-world problems.	Exam prerequisites: None Type of exam: Ongoing tests in the form of course assignments, (10%) and an oral exam, 20 minutes, including everything.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			- Integrals - Infinite series and sequences - Partial derivatives - Multiple integrals - Differential equations	functions. - Calculate and interpret definite and indefinite integrals of real functions. - Perform calculations pertaining to infinite series and sequences. - Solve differential equations.	Communicate and discuss the theory, tools and techniques of calculus.	A selection of approximately 10 of the exercises from the course will form the basis for the exam. During the exam, the student will randomly draw and present one of these exercises. There is no preparation time. The exam will then evolve into a general discussion of the course curriculum. Internal assessment Tools allowed: The student is allowed to bring their notes to the oral exam, but these must be placed on the table during the examination. During the presentation, the student is allowed to consult their notes if they need to, but excessive use of the notes will count negatively towards the grade. During the discussion that follows the presentation, the student is not allowed to consult their notes. Re-exam: Same as ordinary.
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	Permit criteria for attending examination: • Hand in of course project Type of exam: 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-CMC1	Compiler Construction (from A23)	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	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 students will be able to design a small, simple programming language design and implement a compiler for a small, simple programming language using	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			context free grammars list examples of common programming language features	the runtime organization for a programming language	various design patterns and an object oriented language for implementation	The exam is based upon a subject found by draw, and the answers are based on the project produced in the course and handed in according to deadline. Internal assessment. Tools allowed: All Re-exam: Same as the ordinary exam
IT-DCA1	Domain Centric Architecture	5	The student should be able to understand: • Common architectural styles, among other: n-layered, vertical slices, hexagonal, clean, etc. • Different methods for testing • Several architectural patterns • Selected strategic and tactical patterns from Domain Driven Design • Basics of Web API • Basics of Object-Relational Mapping • How to develop a system through automated test	The student should achieve the skills: • Apply architectural patterns in practice to build robust systems • Apply architectural concepts • Split a system by various concerns • Test software using automated tests (unit-/integration-tests). • Design and develop a rich, behaviour-driven domain model, which reflects business-logic • Apply tactical patterns from Domain Driven Design • Effectively use an object-relational mapper for both database-generation and code-scaffolding • Implement an RPC-oriented Web API	The student should be able to: • Implement programs while considering architectural styles, ensuring low coupling, high cohesion, and clear separation of concerns • Discuss different architectural styles, highlighting their strengths, weaknesses • Critically reason about and practically implement key architectural patterns, understanding their implications and benefits • Evaluate and reason about different Web API patterns • Reason about dependencies between system-components • Thoroughly analyze complex domains to develop rich domain models that accurately represent business logic and requirements	
IT-DIM1	Digital Multi Media	5			Having completed this course, students should have profound knowledge of: • Computer Graphics • Design Principles for multimedia • Video, Animation and Sound • XML and Multimedia	Evaluation is based on a written group course assignment, where it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each group member must also hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment.
IT-DIM1	Digital Multi	5			Having completed this course,	Exam prerequisites:

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
	Media (from A23)				students should have profound knowledge of: • Computer Graphics • Design Principles for multimedia • Video, Animation and Sound • XML and Multimedia	None Type of exam: Written exam in the form of a course assignment, handed in before deadline. If the course assignment has been carried out as group work, it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each student must hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam (individually or group). Students who fail the ordinary exam will be given a new deadline to hand in.
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	The course is evaluated via an oral exam after course completion. Grading will be done according to the 7-scale, using an internal examiner. Type of exam: 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.
IT-DSP1	Digital Signal Processing	5	After successfully completing the course, the student will have gained	After successfully completing the course, the student will be able to:	After successfully completing the course, the student will have	Exam prerequisites: None

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
	(from A23)		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	- Record digital signals - Applying different filters (high-pass, low-pass, band-pass, notch) to remove unwanted components of digital signals - Use the Fast Fourier Transform to analyze the frequency content of a signal	acquired competences in: - Explain sampling processes and how to determine the correct sampling frequency - Describe signal processing applications - Applying digital signal processing methods to analyze and interpret engineering problems - Develop signal processing algorithms	Exam type: Individual oral exam, 20 minutes. Exam is based upon an assignment handed in before deadline. The students will present the assignment in the form of a demonstration, followed by questions about the signal processing and feature extraction methods as well as the MATLAB programming. Internal assessment Tools allowed: N/A Re-exam: Same as ordinary exam (new assignment).
IT-EOS1	Embedded Operating Systems	5	Having completed this course, students should be able to Account for advantages and disadvantages of Linux as operating system in embedded systems. Describe the anatomy of a 32-bit embedded system. Describe the features of a Beagle Bone system. 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.	Having completed this course, students should be able to Use basic Linux commands and utilities. Select, install, configure and use tools needed for developing embedded systems. Execute a firmware upgrade on a Beagle Bone system. Install and configure “off the shelf” software in Linux. 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.	Having completed this course, students should be able to Implement shell scripts in BASH Design and implement IoT-devices, based on a 32-bit MCU platform with Linux	Permit criteria for attending examination Mandatory assignments handed in before deadline and accepted. Type of exam: Oral Examination Individual oral examination based upon a subject found by draw. No preparation. Allowed tools: Laptop Internal examiner.
IT-EOS1	Embedded Operating Systems (from A23)	5	Having completed this course, students should be able to Account for advantages and disadvantages of Linux as operating system in embedded systems. Describe the	Having completed this course, students should be able to Use basic Linux commands and utilities. Select, install, configure and use tools needed for developing	Having completed this course, students should be able to Implement shell scripts in BASH Design and implement IoT-devices, based on a 32-bit MCU platform	Exam prerequisites: None. Type of exam: Individual oral exam, 20 minutes,

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			anatomy of a 32-bit embedded system. Describe the features of a Beagle Bone 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.	embedded systems. Execute a firmware upgrade on a Beagle Bone 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.	with Linux	based upon a subject found by draw and without preparation. Internal assessment Tools allowed: Laptop Course hardware kit Re-exam: Same as ordinary exam.
IT-ERP1	ERP systems SAP ABAP/4 Programming	5	Having completed this course, students will be able to: Understand the ABAP Workbench. Create basic ABAP Programs. Understand the control flow and structures in ABAP	Having completed this course, students will be able to: Create Database with domains, data elements and tables Retrieve Data from the Database with open sql. Develop a simple ABAP Programs with modularization. Develop a DYNPRO with navigation (CRUD-functionality) Develop reports with selection screen, alv-list, etc.	Having completed this course, students will be able to: Use the fundamental concepts of the ABAP programming Language Create simple application programs with user dialogs and database connections. Trace the flow of a program and troubleshoot simple problems. Describe change management for new systems. Use ABAP Workbench and basic ABAP language elements. Explain the relationship and difference between the classical procedural programming model and the object-oriented programming model in ABAP/4. Apply screen flow logic and working with external data. Apply different tools and techniques available to implement dataflow in an ABAP program with database. Design and implement an object-oriented SAP application with a database and ALV Grid.	Internal examination The evaluation of the course is based on mandatory course work (50%) and the oral exam (50%) at the end of the course. Only students with approved course work will be allowed to attend the exam. Type of exam: 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.
IT-ERP1	ERP systems SAP ABAP/4 Programming (from A23)	5	Having completed this course, students will be able to: Understand the ABAP Workbench. Create basic ABAP Programs.	Having completed this course, students will be able to: Create Database with domains, data elements and tables	Having completed this course, students will be able to: Use the fundamental concepts of the ABAP programming Language	Exam prerequisites: None Type of exam:

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			Understand the control flow and structures in ABAP	Retrieve Data from the Database with open sql. Develop a simple ABAP Programs with modularization. Develop a DYNPRO with navigation (CRUD-functionality) Develop reports with selection screen, alv-list, etc.	Create simple application programs with user dialogs and database connections. Trace the flow of a program and troubleshoot simple problems. Describe change management for new systems. Use ABAP Workbench and basic ABAP language elements. Explain the relationship and difference between the classical procedural programming model and the object-oriented programming model in ABAP/4. Apply screen flow logic and working with external data. Apply different tools and techniques available to implement dataflow in an ABAP program with database. Design and implement an object-oriented SAP application with a database and ALV Grid.	Individual oral exam, 20 minutes. The exam is in two parts: First part is a presentation and discussion of selected parts of the course work (which consists of 2 mandatory written course assignments, handed in before deadline). Second part is drawn question from the theory of the course. Internal assessment Tools allowed: N/A Re-exam: Same as ordinary exam. New assignments are accepted.
IT-GMD1	Game Development	5	After successfully completing the course, the student will have gained knowledge about: General game development principles, tools, patterns and best practices Game engines and real-time development platforms Unity, a cross-platform game engine 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	After successfully completing the course, the student will have acquired skills in: Navigating Unity and making use of its various features Creating C# scripts to modify game behaviour Structuring game development projects Deploying applications on various hardware 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	After successfully completing the course, the student will have acquired competences in developing industry standard interactive experiences using Unity. The student will be able to possess the developer position within a multidisciplinary game development pipeline, identifying and 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.	Type of exam: Evaluation is based on a written group course assignment, where it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each group member must also hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				the SOLID design principles in a script-based environment		
IT-GMD1	Game Development (from A23)	5	After successfully completing the course, the student will have gained knowledge about: General game development principles, tools, patterns and best practices Game engines and real-time development platforms Unity, a cross-platform game engine 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	After successfully completing the course, the student will have acquired skills in: Navigating Unity and making use of its various features Creating C# scripts to modify game behaviour Structuring game development projects Deploying applications on various hardware 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	After successfully completing the course, the student will have acquired competences in developing industry standard interactive experiences using Unity. The student will be able to possess the developer position within a multidisciplinary game development pipeline, identifying and 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.	Exam prerequisites: None Type of exam: Written exam in the form of a course assignment, handed in before deadline. If the course assignment has been carried out as group work, it must be clearly marked which sections of the course assignment each group member contributed with. Furthermore, each student must hand in an additional 1-2 pages of individual reflections on the work they have done in the course assignment. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam (individually or group). Students who fail the ordinary exam will be given a new deadline to hand in.
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.	Permit criteria for attending examination: Mandatory assignments handed in before deadline and accepted. Type of exam: Oral Examination Individual oral examination (20 min. in total) based upon a subject found by draw. No preparation. Allowed tools: Laptop Internal examiner.
IT-HWP1	Hardware Oriented	5	Having completed this course, students should be able to Seek	Having completed this course, students should be able to	Having completed this course, students should be able to Design a	Exam prerequisites: None

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
	programming (from A23)		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.	Implement low-level drivers for digital I/O-Ports Implement low-level drivers for analogue sensors Implement low-level drivers for analogue actuators.	Hardware abstraction Layer Implement low-level drivers for 8-bit microcontrollers.	Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment Tools allowed: Laptop Course hardware kit Re-exam: Same as ordinary exam.
IT-IDX1	Interaction Design	5			Gain skills within interaction design and usability evaluation. You will achieve: knowledge of and experience in 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	Internal examination. The evaluation of the course is based on mandatory course work (50%) and the oral exam (50%) at the end of the course. Only students with approved course work will be allowed to attend the exam. Type of exam: 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.
IT-IDX1	Interaction Design (from A23)	5			Gain skills within interaction design and usability evaluation. You will achieve: knowledge of and experience in 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	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, without preparation. Exam is based on a question from the course syllabus and based on the course assignment. Internal assessment Tools allowed: N/A

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
					between usability evaluation and interaction design in an iterative design process	Re-exam: Same as the ordinary exam.
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	The basis of the theory, simulation and evaluation of the 3 mandatory course assignments report and presentation. Those project assignments will be part of the examination questions. Type of exam: Oral examination based on one mandatory assignment.
IT-MAL1	Introduction to Machine Learning	5	After having successfully completed the course, the student will have gained knowledge about algorithms, 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 PCA - deep learning methods, e.g. neural networks. - clustering methods, e.g. partitional and hierarchal clustering The students must be able to relate critically and reflectively to the above topics; in particular, it is important that they become proficient in selecting the right type of machine learning method for use in a given context.	After having successfully completed the course, the students should be able to apply the algorithms, 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 algorithms as well as for the proposed solution. In addition, they must be able to reflect on the importance of the context in which the solution is included. Specifically, it is expected that after completion of the course the students will be able to: - Understand and apply a number of machine learning algorithms to both unstructured 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 an appropriate context.	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 students must participate in a group assignment which will also constitute the foundation for the exam. While the assignment is not directly part of the final grade, it will have an indirect influence since the assignment will heavily affect the exam. Each student must explicitly state which sections of the assignment that the student is responsible for, e.g. in the table of contents. If a student does not participate in the assignment, the student will not be able to attend the exam. Type of exam: The course is evaluated based on a 25 minutes oral examination. The examination will depart from the parts of the group assignment that the student is responsible for, and will evolve into a discussion of the syllabus in general. The oral examination does not include a presentation, meaning that the student does not prepare or present a presentation. The student is given one grade

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						based on the examination.
IT-MAL1	Introduction to Machine Learning (from A23)	5	After having successfully completed the course, the student will have gained knowledge about algorithms, 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 PCA - deep learning methods, e.g. neural networks. - clustering methods, e.g. partitional and hierarchal clustering The students must be able to relate critically and reflectively to the above topics; in particular, it is important that they become proficient in selecting the right type of machine learning method for use in a given context.	After having successfully completed the course, the students should be able to apply the algorithms, 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 algorithms as well as for the proposed solution. In addition, they must be able to reflect on the importance of the context in which the solution is included. Specifically, it is expected that after completion of the course the students will be able to: - Understand and apply a number of machine learning algorithms to both unstructured 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 an appropriate context.	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.	Exam prerequisites: At the end of the course, the student must upload a 1-page summary of each of their 6 assignments as well as a 2-page summary of their group project. The summaries must include a brief description of: 1) the assignment problem 2) how the assignment was solved, e.g., data acquisition, data preparation, feature engineering, feature extraction, etc. 3) the algorithms that were used to solve the problem. 4) the performance of the final model 5) a reflection of the learning outcome of solving the assignment. Type of exam: The exam is a 20-minute oral examination that departs from one of the six assignments that the student made during the semester. The exam will also include an examination of the group project report. The final grade will be based on an overall assessment of the six assignments, the group project report, and the oral examination. Internal assessment Tools allowed: N/A Re-exam: Same as ordinary exam.
IT-MAL1	Introduction to Machine Learning and AI (From S24)	5	After having successfully completed the course, the student will have gained knowledge about algorithms, methods, techniques, tools, and applications within the following	After successfully completing the course, the student will have developed the following skills: • Ability to prepare	Upon completion of the course, students are expected to have acquired the competences to: • Make informed decisions regarding the selection and	Exam prerequisites: At the end of the course, the student must upload a 1-page summary of each of their 6 assignments as well as a 2-page summary of their group

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			machine learning methods: • Different data preparation and preprocessing methods • Different types of classification algorithms, e.g., Naïve Bayes, k-Nearest Neighbor, Decision Trees, Logistic Regression, Support Vector Machines • Different types of regression algorithms, e.g., simple linear regression, multiple linear regression, Ridge regression, Lasso regression • Different types of dimensionality reduction algorithms, e.g., Principal component analyses, singular value decomposition, factor analysis • Different types of clustering algorithms, e.g., k-Means clustering, Agglomerative clustering, DBSCAN • Different metrics for assessing the strength and quality of their machine learning algorithms	and preprocess data for various machine learning applications effectively. • Proficiency in implementing and tuning classification algorithms and selecting the appropriate classifier for a given dataset. • Capability to apply regression techniques to predict continuous variables and evaluate the predictive ability of regression models. • Using dimensionality reduction algorithms to interpret and simplify complex datasets. • implementing clustering algorithms to categorize unlabelled datasets and determining the optimal number of clusters. • Ability to use various machine learning tools and libraries • Critical evaluation of model performance using various metrics and validation techniques.	application of machine learning techniques tailored to specific problem domains. • Fine-tune and parametrize machine learning algorithms to optimize their performance on specific datasets. • Conceptualize, design, and develop machine learning solutions for real-world problems. • Articulate, communicate, and deliberate machine learning solutions, their implications, and associated decisions with both domain experts and non-technical persons.	project. The summaries must include a brief description of: 1) the assignment problem 2) how the assignment was solved, e.g., data acquisition, data preparation, feature engineering, feature extraction, etc. 3) the algorithms that were used to solve the problem. 4) the performance of the final model 5) a reflection of the learning outcome of solving the assignment. Type of exam: The exam is a 20-minute oral examination that departs from one of the six assignments that the student made during the semester. The exam will also include an examination of the group project report. The final grade will be based on an overall assessment of the six assignments, the group project report, and the oral examination. Internal assessment. Tools allowed: N/A Re-exam: Same as the ordinary exam.
IT-NSQ1	No-SQL versus relational databases	5	The student should be able to • describe document-based and graph databases • explain updating and querying in different database paradigms • explain updating and querying in GraphQL • explain schemas and constraints in non-relational databases • compare relational and different	The student should be able to apply • modelling techniques in document-based and graph databases • schemas and constraints to enforce designs in a no-SQL database • APIs and languages to maintain and query databases • setting up No-SQL databases in	At the end of the course, the students should be able to • make an informed choice of database management system • design and create a data model in the chosen database system • set up a cloud environment to use for the data model	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 exam

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			non-relational approaches to database design.	the cloud		
IT-NSQ1	No-SQL versus relational databases (from A23)	5	The student should be able to • describe document-based and graph databases • explain updating and querying in different database paradigms • explain updating and querying in GraphQL • explain schemas and constraints in non-relational databases • compare relational and different non-relational approaches to database design.	The student should be able to apply • modelling techniques in document-based and graph databases • schemas and constraints to enforce designs in a no-SQL database • APIs and languages to maintain and query databases • setting up No-SQL databases in the cloud	At the end of the course, the students should be able to • make an informed choice of database management system • design and create a data model in the chosen database system • set up a cloud environment to use for the data model	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes without preparation. Exam is based upon two course assignments handed in before deadline, and it is covering mandatory course work and theory covered in the course. Internal assessment Tools allowed: N/A Re-exam: Same as ordinary exam
IT-PCL1	Programming Concepts and Languages	5	Having successfully completed this course, the student should be able to: - describe the key concepts and have a basic understanding of different programming paradigms and languages. - understand and use the functional paradigm. - develop small and medium size programs/apps using F# and Python programming languages. - Understand and use the Python language with various middleware such as Django, Flask, and RabbitMQ	Having successfully completed this course, the student should be able to: - use the different programming paradigms to solve a particular programming problem. - Use one of the Web Frameworks to develop a distributed application.		The final examination is a three-hour written examination. Internal examination.
IT-PCL1	Programming Concepts and Languages (from A23)	5	Having successfully completed this course, the student should be able to: - describe the key concepts and have a basic understanding of different programming paradigms and languages. - understand and use the functional paradigm. - develop small and medium size	Having successfully completed this course, the student should be able to: - use the different programming paradigms to solve a particular programming problem. - Use one of the Web Frameworks to develop a distributed application.		Exam prerequisites: None Type of exam: Individual written exam, 3 hours Internal assessment Tools allowed: All aids are allowed included access to online material. However,

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			programs/apps using F# and Python programming languages. - Understand and use the Python language with various middleware such as Django, Flask, and RabbitMQ			it is not allowed to use AI tools such as ChatGPT and similar AI and Machine Learning driven tools and chatbots. Re-exam: Same as the ordinary exam.
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: Requirement Specification - IEEE 830 standard" document for a project Test Specification - IEEE 829 standard" document for a project Project relations to CMMI model" document for a project.	The course is assessed on the basis of 3 individual assignments, weighing 25%, 30% and 45%, respectively. If the course is failed, the student must go for internal oral re-examination.
IT-PME1	Process Management for ICT Engineering (from A23)	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: Requirement Specification - IEEE 830 standard" document for a project Test Specification - IEEE 829 standard" document for a project Project relations to CMMI model" document for a project.	Exam prerequisites: None Type of exam: Ongoing assessment in the form of two group assignments of max two persons (assignment 1 and 2) and one individual assignment (assignment 3), weighing 25%, 30% and 45 % respectively. All assignment must have a grade of at least 02 and the final grade is a weighted grade according to the assignments individual weight. Internal assessment Tools allowed: N/A Re-exam: Individual oral re-exam, 20 minutes Internal assessment.
IT-RTP1	Real-Time Programming, Interfacing and	5	Having completed this course, the student has gained knowledge in the below areas. Specifically, the	Having completed this course, the student should be able to: Write functioning real-time programs in C	Having completed this course, students should be able to: master and use simple real-time operating	Permit criteria for attending examination: Mandatory assignment handed in before

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
	Electronics		student is able to: Understand the basic concepts of real-time programming Explain issues like deadlocks, priority inversion etc.	using FreeRTOS Analyse a simple real-time design for schedulability, deadlocks, utilization etc.	systems be able to analyse/design/describe and construct real-time programs understand timers and clocks, and how they are used in real-time programming understand synchronization avoiding dead-locks and priority inversion understand memory management, resource sharing and control be able to design and construct real-time systems using FreeRTOS and C-programming understand low-level protocols, CRC etc.	deadline and accepted. Type of exam: Oral examination based on mandatory assignment.
IT-RTP1	Real-Time Programming, Interfacing and Electronics (from A23)	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-time systems using FreeRTOS and C-programming understand low-level protocols, CRC etc.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, based upon a subject found by draw and without preparation. Internal assessment Tools allowed: Laptop Re-exam: Same as ordinary exam.
IT-SCP1	IT Security and Cryptography in Practice	5	After successfully completing the course, the student will have gained an understanding of the cross-disciplinary nature of cyber security, and the complexities, challenges and wider implications of the contexts in which cyber security problems occur in the workplace. knowledge about several key implementations of cryptography and other IT-security related issues.	After successfully completing the course, the student will be able to Draw on and apply relevant IT security approaches, tools and frameworks for IT security enquiry to different settings in real world situations. Frame and address IT security problems, questions and issues as an IT security project, being aware of the environment and context in which the problem exists.	After successfully completing the course, the student will have acquired competences in Applying complex cryptographic primitives to real-world cases Document and explain an IT-security project clearly and unambiguously to peers Review, evaluate and reflect upon knowledge, skills and practices in cyber security.	The student must attend three mandatory seminars: i) introduction, ii) midway, iii) final. The student must hand in two compulsory papers: i) midway paper, ii) final paper. Students are assessed by the lecturer based on 1) Final paper (50%) 2) Participation and performance at final seminar (20%) 3) Midway paper (10%)

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						4) Participation and performance at midway seminar (20%) If a student fails to meet one or more of the above requirements for passing the course, the student will be given an extra assignment whose scope depends on the scope of the missing requirements.
IT-SCP1	IT Security and Cryptography in Practice (from A23)	5	After successfully completing the course, the student will have gained an understanding of the cross-disciplinary nature of cyber security, and the complexities, challenges and wider implications of the contexts in which cyber security problems occur in the workplace. knowledge about several key implementations of cryptography and other IT-security related issues.	After successfully completing the course, the student will be able to Draw on and apply relevant IT security approaches, tools and frameworks for IT security enquiry to different settings in real world situations. Frame and address IT security problems, questions and issues as an IT security project, being aware of the environment and context in which the problem exists.	After successfully completing the course, the student will have acquired competences in Applying complex cryptographic primitives to real-world cases Document and explain an IT-security project clearly and unambiguously to peers Review, evaluate and reflect upon knowledge, skills and practices in cyber security.	Exam prerequisites: 3 mandatory assignments handed in: 1) A 1-page summary of their project idea. 2) A 1-page summary of their midterm seminar report. 3) A 1-page summary of their final report. If a student fails to meet one or more of the above mandatory assignments, the student will be given an extra assignment, to qualify for re-exam. The scope of this assignment depends on the scope of the missing requirements. Type of exam: The exam has ongoing assessment. Midway exam based on Midway Paper (30%) Final Exam based on Final Paper (70%) Internal assessment Tools allowed: All Re-exam: The re-exam consists of two parts: 1) A 1-page summary of each of the main topics in the course, incl. the student's own topic (10%) 2) A 20-minute oral examination based on Final Paper (90%)

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						The student may choose to resubmit a revised version of the final paper. The main topics of the course are determined by the students at the beginning of the course and consists of the topics of their final paper.
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	Grading will be done according to the 7-scale, using an internal examiner. Type of exam: 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. The re-exam may be held as an oral examination.
IT-SMP1	Stochastic Modelling and Processing (from S24)	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	Exam prerequisites: None The final exam has two parts. • The first part is a Flowlock exam in Wiseflow. • The second part is a Wiseflow exam without Flowlock. The second part must be completed in the Jupyter Notebook environment and the answers must be submitted in Wiseflow. The exam has a total duration of 4 hours. The student will not be able to access the second part before the first part is concluded. Each part has an equal weight in the final grade. Internal assessment

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
						Tools allowed: In the first part the students are allowed to use any notes, books, and/or other written/printed material and will have access to pdf files on their laptop. The student may bring their own calculator. In the second part all supplementary materials and aids are allowed, e.g., using a computer as a reference work. It is not allowed, however, to use AI-tools such as CoPilot, ChatGPT, Bing, etc. as per the general VIA rules. Communication of any sort is not allowed during the exam and will lead to expulsion of all involved parties from the exam. Re-exam: Re-exams may be oral.
IT-SWA1	Single-Page Web Applications	5	The student should be able to - describe dynamic and static languages - apply common design patterns to document object models - explain techniques and pitfalls of asynchronous programming - explain the prototype model and contrast it with class-based inheritance - compare single-threaded and multi-threaded asynchronous model - explain the elements of the TypeScript type system	The student should be able to apply - Object-oriented programming in JavaScript and TypeScript - Functional programming in TypeScript - Callbacks and higher-order functions in TypeScript - Asynchronous programming using Promises - Manipulating web pages using JavaScript and TypeScript - Calling web services using XmlHttpRequest	At the end of the course, the students should be able to - design and construct single-page application using AJAX techniques, including JavaScript, TypeScript, HTML DOM, XML, JSON and web services.	Oral examination based on a question from the course syllabus based on one of the course assignments. Approximately 20 minutes (including assessment) without preparation. Internal exam
IT-SWA1	Single-Page Web Applications (from A23)	5	The student should be able to - describe dynamic and static languages - apply common design patterns to document object models - explain techniques and pitfalls of asynchronous programming - explain the prototype model and contrast it with class-based	The student should be able to apply - Object-oriented programming in JavaScript and TypeScript - Functional programming in TypeScript - Callbacks and higher-order functions in TypeScript - Asynchronous programming using Promises	At the end of the course, the students should be able to - design and construct single-page application using AJAX techniques, including JavaScript, TypeScript, HTML DOM, XML, JSON and web services.	Exam prerequisites: None Type of exam: Individual oral exam, 20 minutes, without preparation. Exam is based on a question from the course syllabus and based on one or more of the course

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			inheritance - compare single-threaded and multi-threaded asynchronous model - explain the elements of the TypeScript type system	- Manipulating web pages using JavaScript and TypeScript - Calling web services using XmlHttpRequest		assignments. Internal assessment Tools allowed: N/A Re-exam: Same as the ordinary exam.
IT-XRD1	XR Development	5	After successfully completing the course, the student will have gained knowledge about: - XR, AR, VR, AV & MR terminology - State of the art and the evolution of AR & VR hardware - Use cases for AR and VR applications - Tracking technologies for XR - Display technologies for XR - The Vuforia Engine - ARCore and ARFoundation - XR Interaction Toolkit - The Unity XR tech stack & OpenXR - XR Interaction techniques - Rendering challenges in XR - Spatialized audio - OVR and similar integrations for hand tracking, locomotion and specialized interactions	After successfully completing the course, the student will have acquired the skills to: - Analyse and optimize an AR or VR development workflow - Compare and utilize various SDK offerings and libraries for XR development - Work with and reflect on the theory behind prominent challenges in the XR industry such as tracking, rendering, locomotion and input - Reflect on underlying sensor and display technologies for XR hardware - Classify XR applications and reflect on their use cases - Describe, compare and apply various interaction techniques in XR	After successfully completing the course, the student will have acquired competencies to develop marker based and markerless augmented reality applications and mobile virtual reality applications.	Prerequisites: If the compulsory projects and course activities are not approved by the lecturer, the student will be denied access to the exam and will thus have used one exam attempt. Type of exam: Written assignment spanning the semester. Assessment is based on projects developed in groups where it must be clearly marked which parts of the applications each group member contributed to. Furthermore, each group member must hand in an additional 1-2 pages (one page is considered 2400 characters) of individual reflections on the work they have done. Assessed by the lecturer and an internal assessor. Allowed tools: NA Re-exams: Students who fail the ordinary exam will be given a new deadline for handing in a revised assignment.
IT-XRD1	XR Development (from A23)	5	After successfully completing the course, the student will have gained knowledge about: - XR, AR, VR, AV & MR terminology - State of the art and the evolution of AR & VR hardware	After successfully completing the course, the student will have acquired the skills to: - Analyse and optimize an AR or VR development workflow - Compare and utilize various SDK offerings and libraries for XR	After successfully completing the course, the student will have acquired competencies to develop marker based and markerless augmented reality applications and mobile virtual reality applications.	Exam prerequisites: None Type of exam: Written assignment spanning the semester. Assessment is based on projects

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			- Use cases for AR and VR applications - Tracking technologies for XR - Display technologies for XR - ARCore and ARFoundation - XR Interaction Toolkit - The Unity XR tech stack & OpenXR - XR Interaction techniques - Rendering challenges in XR - Spatialized audio - OVR and similar integrations for hand tracking, locomotion and specialized interactions	development - Work with and reflect on the theory behind prominent challenges in the XR industry such as tracking, rendering, locomotion and input - Reflect on underlying sensor and display technologies for XR hardware - Classify XR applications and reflect on their use cases - Describe, compare and apply various interaction techniques in XR		developed in groups where it must be clearly marked which parts of the applications each group member contributed to. Furthermore, each group member must hand in an additional 1-2 pages (one page is considered 2400 characters) of individual reflections on the work they have done. Assessed by the lecturer and an internal assessor without oral presentation or defence. Tools allowed: N/A Re-exam: Same as the ordinary exam (individually or group assignment if possible). Students who fail the ordinary exam will be given a new deadline to hand in.
ME-MAT1	Mathematics 1	5	After completing the course, the student can: Explain limits of simple expressions Describe the meaning of a function's derivative Reproduce rules for derivatives, including the chain rule Explain the meaning of partial derivatives Explain the connection between definite integrals and areas under and between graphs Identify order and type of ordinary differential equations Explain the use of polar coordinates Explain how the complex numbers is an extension of the reals	After completing the course, the student can: Find limits Calculate derivatives using standard differentiation rules Calculate partial derivatives Determine characteristics of curves, including tangent vector, normal vector and curvature Calculate indefinite integrals, using substitution and integration by parts when appropriate Determine definite integrals Convert between Cartesian and polar coordinates Apply complex numbers, including conversion between different representations	After completing the course, the student must be able to: Use the covered methods in other courses when appropriate. Read texts that use the notation and concepts covered.	Prerequisites for exam: None Exam type: Written exam Duration is 4 hours Grade is exclusively based on the exam External censor Tools allowed: All usual, but no communication and no use of the web during the exam Re-exam: Same as ordinary
SE-LCA1	Circular Economy and LCA	5	Students completing this course will be familiar with: The international guidelines for LCA analyses (ISO standards 14040 and 14044). The step-by-step working process that must be followed when carrying out	Students completing this course will be able to: Define functional units, system boundaries and time scopes for LCA analyses according to the guidelines. Carry out LCA analyses for simple production or	Students completing this course will be able to: Define comparable scenarios for competing production/service systems in order to analyse the respective environmental impacts of these	Prerequisites: Mandatory course activities completed. Mandatory assignments handed in before deadline and accepted.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			an LCA analysis. The principles behind defining functional units, system boundaries and time scopes for LCA analyses. Chosen data sources providing data for LCI´s and LCIA´s. Different environmental impact categories. The common way to graphically present end results of LCA analyses. How the UN system influences global development within CE. The UN SGDs	service system scenarios according to the guidelines. Compare competing production or service systems based on an LCA analysis. Present and interpret results of LCA analyses and discuss these in relation to decision-making. Search for and identify relevant data for Life Cycle Inventories (LCI). Prepare simple Life Cycle Inventories (LCI) and carry out Life Cycle Impact Assessments (LCIA) based on these, according to the guidelines. Graphically present the results of LCA analyses and explain how these are related to the former steps of the analyses. Carry out an LCA by using the program "LCABYG". Identify barriers to change of CE development. Identify opportunities for CE business development. Make a simpel business model. Formulate individual change of behaviour to promote CE. Evaluate business cases in relation to fulfilling the SDG. Promote circular economy as an innovation tool for companies.	Relate results from LCA analyses with the ideas of CE to suggest sustainable choices in given situations Discuss how working towards fulfilling the SDGs requires individual as well as a political change of behaviour Reflection about business models and product development in CE.	Type of examination: A case based written exam with internal examiner. Allowed tools: Re-exam: Not passing the course - a new course assignment will be given, to be accepted and evaluated in equal manner as within the course.
SE-LCA1	Circular Economy and LCA (from A23)	5	Students completing this course will be familiar with: The international guidelines for LCA analyses (ISO standards 14040 and 14044). The step-by-step working process that must be followed when carrying out an LCA analysis. The principles behind defining functional units, system boundaries and time scopes for LCA analyses. Chosen data sources providing data for LCI´s and LCIA´s. Different environmental impact categories. The common way to graphically present end results of LCA analyses. How the UN system influences global development within CE. The UN SGDs	Students completing this course will be able to: Define functional units, system boundaries and time scopes for LCA analyses according to the guidelines. Carry out LCA analyses for simple production or service system scenarios according to the guidelines. Compare competing production or service systems based on an LCA analysis. Present and interpret results of LCA analyses and discuss these in relation to decision-making. Search for and identify relevant data for Life Cycle Inventories (LCI). Prepare simple Life Cycle Inventories (LCI) and carry out Life Cycle Impact Assessments (LCIA)	Students completing this course will be able to: Define comparable scenarios for competing production/service systems in order to analyse the respective environmental impacts of these Relate results from LCA analyses with the ideas of CE to suggest sustainable choices in given situations Discuss how working towards fulfilling the SDGs requires individual as well as a political change of behaviour Reflection about business models and product development in CE.	Prerequisites: None Type of examination: A case based written exam, 48 hours Internal examiner. Allowed tools: Re-exam: Same as the ordinary exam, or re-exams may be oral.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				based on these, according to the guidelines. Graphically present the results of LCA analyses and explain how these are related to the former steps of the analyses. Carry out an LCA by using the program “LCABYG”. Identify barriers to change of CE development. Identify opportunities for CE business development. Make a simpel business model. Formulate individual change of behaviour to promote CE. Evaluate business cases in relation to fulfilling the SDG. Promote circular economy as an innovation tool for companies.		

Appendix 2: Courses for the Software Engineering Programme, semester 4-7 in Viborg

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
IT-AVR1	Augmented and Virtual reality technologies (Viborg)	10	After successfully completing the course, the student will have gained knowledge about: General understanding of what Virtual, Augmented and Mixed reality is. History and past development of VR/AR. Core mechanics in AR/VR from a user point of view. General knowledge about Mobile VR. How does Gear VR work and what are the differences between other mobile VR devices. Core components of Oculus VR SDK. General knowledge about Mobile AR. General understanding of hardware that makes AR possible. In-depth knowledge of Vuforia and how does Vuforia Fusion 7 work. Another AR SDKs, such as ARCore or ARKit. General differences between Camera based AR and Holographically based AR. What is Microsoft HoloLens and how does it work (programmatically and hardware). Core components of Microsoft Mixed reality toolkit related to MS HoloLens. Spatialized Audio and human hearing. Spatialized audio framework FMOD. Reverberation on sound reflection synthetization made with Google Resonance SDK. General understanding of Desktop VR.	After successfully completing the course, the student will have acquired skills in: Choosing appropriate technology for the assigned project. Building VR / AR interactive experiences for various technologies and platforms in Unity 3D. Utilizing various SDKs related to creation of VR/AR experiences. Structuring AR / VR projects. Creating applications with appropriate AR/VR ergonomics. Creating and modifying existing C# scripts used with related SDKs. Applying theories to achieve as deep immersion as possible. Navigating the SDKs documentations. Handling and optimizing performance for taught technologies.	After successfully completing the course, the student will have acquired competences in: Developing industry standard interactive AR / VR experiences using Unity3D.Possessing the developer position within a multidisciplinary AR / VR experience development pipeline. Identifying and executing on the technical requirements of the developed product. Having a solid foundation to further professional skills in AR / VR industry.	Each group is evaluated upon their performance in each assignment respectively using the 7 grade system. Allowed tools: All The course must be passed before 15/6 for spring semester and 15/12 for autumn semester.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			What is and how HTC Vive works. Differences between Oculus Rift and HTC Vive. Core components of Steam VR SDK and VRTK. What is and how LeapMotion works. Core components of Leap. Motion SDK. Advantages of technology / different SDK combinations (i.e. HTC Vive + Leap. Motion or FMOD + Resonance SDK + Vive / HoloLens).			
IT-AVR2	Advanced Augmented and Virtual Reality Technologies (Viborg)	10	General understanding of market, development workflows, deployment and publishing for AR applications What is and what are the pros and cons of cloud-based systems for storage and processing Understanding of structured data formats Understanding of relational databases and query building Understanding render pipelines and how to choose between them Understanding shaders visually and how they work programmatically	Optimization, web solutions and AR native device support Choosing appropriate technology for the assigned project Creating native AR applications by utilizing SDKs Utilizing spatial data and spatial awareness in AR/MR-applications Testing cross-platform interactive applications Utilizing events to interact with integrated web applications Creating cross-platform applications that communicate with a core-platform backbone Web & Cloud integration with Databases Setting up a file storage system in a cloud-based system Uploading and utilizing files in the cloud Utilizing cloud processing for analysis of cloud data Communicating with servers in the cloud using signals and events from multi-platform applications Analyzing and querying relational databases within VR and AR applications	Developing industry standard AVR applications storage and/or processing in a cloud-based system Developing industry standard cross-platform AVR applications To possess the position of graphical programmer in an AVR development team Optimizing AVR applications	Oral Examination Individual oral examination based upon a subject found by draw with 20 minutes preparation time. All tools are allowed during preparation. External examiner The course must be passed before 31st of January for autumn and before 31st of August for spring.

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				Shaders and Rendering Choosing an appropriate render pipeline for the assigned project Building shaders visually Programming basic shaders		
IT-AVR3	Augmented and Virtual reality technologies (Viborg)	5	After successfully completing the course, the student will have gained knowledge about: General and specific understanding of motion capture. Understanding of advanced technology for AVR platforms. Understand how to utilize API to develop for advanced technologies.	After successfully completing the course, the student will have acquired skills in: How to use a Motion Capture system How to setup a suit for recording data How to use the recording system for motion capture data How to clean up data How to choose a relevant data format for the data and for a game-engine How to import and setup the data in a game-engine How to develop for a chosen example of an advanced technology How to understand and use a specialized hardware/technology using an API How to implement an AVR prototype based on the technology and the API	After successfully completing the course, the student will have acquired competences in: To be more effective working in teams by using API's for developing solutions. As a developer to understand how to use data in motion capture systems and what is expected as a developer in a team. To understand and use communication of different activities in the project. Have a solid understanding of specialized AVR hardware in AR / VR industry.	The examination is an external oral examination based on first a group presentation for the project. Each member must present at least one page. 10 minutes per group. Then an individual examination for 20 minutes. The student first presents his/her role(s) in one of the pro-jects and the code created. This is 5 minutes. Then 15 minutes of discussion and questions. The time is including evaluation and grading. Examinations account for 100 % of final grade. Allowed tools: All. Deadlines for passing the course: The course must be passed before 30/8 for spring semester and 31/1 for autumn semester. Re-exam will be possible in connection with the following ordinary exam.
IT-BPR1	Bachelor Project 1 (Viborg)	5	After successfully completing the course, the student will have gained knowledge about: Performing information search and retrieval Project planning Delimitating a problem and a problem domain Motivation theory	After successfully completing the course, the student will have acquired skills in: Identify and justify problems and their context Select and argue for choice of methods and reflect critically on said methods Find and assess relevant literature	After successfully completing the course, the student will have acquired competences in: Describe and delimit a larger Software Engineering project Plan and structure the project within the set time limit Initiate the preliminary steps in a systems development process, leading to a clearly defined	Oral Internal assessment The basis of the evaluation is the "Project description" and a draft of the analysis part of the project. Pass/fail based on the quality of the handed in work. Additional information The Project Description must contain minimum 8 pages plus

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			Team work and team dynamics Communication and presentation skills Testing	within the problem domain Present the result for an audience of engineers	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	appendices. The analysis part of the project must contain at least requirements, use case and descriptions, activity diagrams, analysis class diagram with a scope of minimum 5 pages. Allowed tools: All.
IT-BPR2	Bachelor Project 2 (Viborg)	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 Software Engineering Project Select and use relevant theories and methods to solve the problem Plan and structure the project within the BPR2 time frame Initiate the preliminary steps in a system development process, leading to a clearly defined requirements capture, use cases as well as object and behaviour analysis. Work successfully in a project group with the objective of solving a well-defined engineering problem.	Oral examination. 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.
IT-DAP1	Digital Animation production (Viborg)	5	After successfully completing the course, the student will have gained knowledge about: General understanding of where 3D is used in AR/VR production. The production pipeline for 3D Assets.	After successfully completing the course, the student will have acquired skills in: Familiarity of the 3D software UI and basic settings and functions Understanding of working in hierarchies Setting up a project Understanding file structures in projects Basic polygon modelling as a tool for modelling for 3D engines How to export models to 3D	After successfully completing the course, the student will have acquired competences in: Developing industry standard 3D models using a 3D tool. To understand the connection between the positions as a developer and 3D modeler, within a multidisciplinary AR / VR experience development pipeline Have a foundation to further professional skills in AR / VR industry.	Type of examination: 7 grade evaluation of the total number of assignments. The evaluation is the total of all handed-in assignments. Course assignments account for 100 % of final grade

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				engines. Understanding basic UV workflows. Creating textures and applying them to the geometry Understanding shaders as a basic for textures Creating simple environments using polygon modelling Basic object - and camera animation Basic rendering		
IT-GMD2	Advanced Game Development (Viborg)	5	After successfully completing the course, the student will have gained knowledge about: Useful alterations to common design patterns Chosen design patterns that are common in game development How a game engine is structured and implemented Asymptomatic notation for running time and space requirements Different linear data structures (sets, maps, lists and stacks) Different algorithm types and templates Different sorting and searching algorithms	After successfully completing the course, the student will have acquired skills in: Utilizing design patterns when developing games Reusing and modularizing code Being able to analyse time and space complexity of an algorithm Choosing the appropriate data structure for a problem Being able to design and implement algorithms in an object-oriented language Being able to design and implement data structures in an object-oriented language	After successfully completing the course, the student will have acquired competencies in developing large maintainable applications. The student will be able to take the roles of developer and software architect in multidisciplinary game development teams. The student will also be able to analyse and improve algorithms and implementations.	Oral examination Individual oral examination based upon a subject found by draw with 20 minutes preparation time. All tools are allowed during preparation. External examiner
IT-INP1	Engineering Internship (ICT)	30	The student must: gain knowledge of theory, methodology and practice within a profession or one or more fields of study be able to understand and reflect on theories, methodology and practice be aware of non-technical – societal, health and safety, environmental, economic and industrial – implications of engineering practice	The student must: be able to apply the methodologies and tools of one or more fields of study and to apply skills related to work within the field/fields of study or profession be able to assess theoretical and practical problems and to substantiate and select relevant solutions be able to communicate professional issues	The student must: be able to handle complex and development-oriented situations in study or work contexts be able to independently participate in professional and interdisciplinary collaboration with a professional approach be able to identify own learning needs and to organise own learning in different learning environments promote an engineering-oriented approach during the remaining semesters on the Bachelor programme	

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
					develop personal skills required for the professional career as engineer form the basis for developing personal/professional network.	
IT-PCO1	Physical Computing (Viborg)	5	After successfully completing the course, the student will have gained knowledge about: More in depth working knowledge about the process of printing and cutting. The importance of planning to avoid bottle necking with printing and cutting. Knowing about possibilities in different materials. Knowledge about how to use electronics and sensors.	After successfully completing the course, the student will have acquired skills in: Creating and using 3D print and laser cutting. How to setup electronics and sensors. How to develop different projects using Arduino and relevant sensors. How to develop embedded code in Arduino. How to input the data into interactive applications. How to act in interactive applications based on input.	After successfully completing the course, the student will develop competence to: Importance of making decisions in order to move forward in production. Working with advanced functionality. Set and meet deadlines based on a structured working process. Understand how as a software engineer, you can use 3D print and laser cutting to make interactive applications. Being able to create interactive applications with use of electronics and sensors.	Permit criteria for attending examination: Course assignment handed in before deadline Oral examination The examination is an oral examination based on a presentation of the project. 5 minutes per member/group. Then an individual examination for 20 minutes. The student first presents his/her interactive applications, 3D print and laser cut in the project and the code created. This is 5 minutes. Then 15 minutes of discussion and questions and also including evaluation and grading. Allowed tools: All The course must be passed before 31/8 for spring semester and 31/1 for autumn semester.
IT-SEP4V	Semester Project 4V (Viborg)	10	After successfully completing the course, the student will have gained knowledge about: General user experience principles, tools, patterns and best practices What kind of VR/AR devices and equipment exist Identifying users Knowledge about good User Experience practices Knowledge about bad User Experience practices How to test User Experience How to prototype	After successfully completing the course, the student will have acquired skills in: Creating user profiles Analyzing the User Experience of an application Designing User Experience tests Creating a usability document Detecting the locomotion of an VR/AR application Creating prototypes Gather information about the needs of target/persona users	After successfully completing the course, the student will have acquired competences in developing creating a better user experience when designing a product. The student will be able to possess the UX designer role within a multidisciplinary application development pipeline, identifying User experience problems and come up with solutions to said problems.	Oral Examination The students hand-in a report. The report must be 5 pages for the group and then extra 5 pages maximum for each student in the group. So, a group of four (4) students must hand in maximum 5+(4x5) = 25 pages. This is a maximum page number and more pages will reduce the grade. The report is 25 % of the exam. At exam the group starts by

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
			Locomotion of VR applications The importance of conducting user research	Apply the fundamentals of user experience design		presenting their project using visual and a structured method like PowerPoints, images, video and prototypes. This is 5 minutes for each member of the group. This is 25 % of exam. Then there are 5 minutes group presentation for the product. This is 25 % of exam. Finally, there is 15 minutes individually student examination. This is 25 % of exam. Re-examination: Any re-exam will take place in the same way as the ordinary exam. External Examiner. Allowed tools: All. The course must be passed before 15/6 for spring semester and 15/12 for autumn semester.
IT-SEP6	Semester Project 6 - Viborg	10	After successfully completing the course, the student will have gained knowledge about: More in depth working processes for production and planning The importance of planning to avoid bottle necking Experience using folder structure, pipeline and workflow Experience in following the direction of the project Team management	After successfully completing the course, the student will have acquired skills in: Adapting their pipeline, folder structure and production plan from the given standards Work successfully as a team with them Give estimates of their worktime and track their progress Present their project as presentations How to write an analysis of the project and a reflection on their work Implementation of projects	After successfully completing the course, the student will develop competence to: Importance of making decisions in order to move forward in production Working with advanced functionality Set and meet deadlines based on a structured working process Create, analyse and edit as needed the teams group work protocol and their role Collaborate, delegate and communicate clearly within a group	Before exam the examiner and censor have read the report and the process-report. The handed-in product is also checked. The evaluation at exam is then that report, process report, product, presentation at exam and answers at exam are evaluated in an equal manner with 20% for each. The examination is an oral examination based on first a group presentation for the project. Each member must present at least one page. 10 minutes per group. Then an individual examination for 20 minutes. The student first presents his/her role(s) in the project and the code created. This

CODE	TITLE	ECTS	KNOWLEDGE	SKILLS	COMPETENCES	EXAMINATION
				Communication with team members Create and document tests for the developed products.		is 5 minutes. Then 15 minutes of discussion and questions and also including evaluation and grading. Allowed tools: All Please note that re-examinations may take a different form than the ordinary exams. The course must be passed before 31/8 for spring semester and 31/1 for autumn semester.
IT-AUD1	Advanced Unity Development (Viborg)	5	After successfully completing the course, the student will have gained knowledge about: Data Oriented Programming Unity's Job System Unity's Entity Component System Unity's Burst Compiler Unity ECS Authoring Lambda Expressions in C# The LINQ library for C#	After successfully completing the course, the student will have acquired skills in: Designing a Data Oriented structure for your game or interactive experience Implementing a game or experience using the Data Oriented Tech Stack in Unity Writing and understanding LINQ queries	After successfully completing the course, the student will be able to take the role of programmer in multi-disciplinary game development teams working on performance critical code.	Oral Examination Individual oral examination based upon a subject found by draw. There is 20 minutes preparation time and 20 minutes examination including grading. Expect 16 to 18 minutes of examination and 2 to 4 minutes of grading. Allowed tools: All tools are allowed during preparation. Internal examiner
IT-UEX1	User Experience (Viborg)	5	After successfully completing the course, the student will have gained knowledge about: General user experience principles, tools, patterns and best practices Identifying users Knowledge about good User Experience practices Knowledge about bad User Experience practices How to test User Experience How to prototype Locomotion of VR applications The importance of conducting user research	After successfully completing the course, the student will have acquired skills in: Creating user profiles Analyzing the User Experience of an application Designing User Experience tests Creating a usability document Detecting the locomotion of an VR/AR application Creating prototypes Gather information about the needs of target/persona users Apply the fundamentals of user experience design	After successfully completing the course, the student will have acquired competences in developing creating a better user experience when designing a product. The student will be able to possess the UX designer role within a multidisciplinary application development pipeline, identifying User experience problems and come up with solutions to said problems.	Individual, written exam, 24 hours. The students will have 24 hours to create a User Experience document.