Offered by
ICT Engineering

IT-CAL1

1.1

Computer Architecture Lab

ECTS
10

Prerequisites
Knowledge about basic computer programming.

- SDJ1

Main purpose
The main purpose of the course is to provide students with an introduction to computer architecture and low-level software development, including tools and theory needed to write working software for embedded systems (ASM/C).

Knowledge
Having completed this course, the student has gained knowledge in the below areas. Specifically, the student is able to:

- Explain numbering representations, including two’s complement to represent negative numbers in the binary numbering representation
- Understand the functionality of basic logic gates, half- and full-adders, flip/flops, etc.
- Understand how Boolean expressions can be used to reduce Boolean expressions
- Explain the architecture of simple CPUs and how they function, explain the build and working behavior of basic building blocks of CPUs (registers, ALUs, etc.)
- Explain assembler programming
- Explain Clocks, instruction set layout, memory architectures, addressing modes
- Explain basic C programming artefacts and constructs
- Understand pointers and simple data structures in C
- Understand Test Driven Development (TDD)

Skills
Having completed this course, the student should be able to:

- Write functioning C programs for embedded systems
- Use automated tests (Unit tests) to verify behaviour of C programs
- Use UML to document C programs
- Write functioning assembler programs for microcontrollers
- Calculate execution time of ASM programs (AVR MCU)
- Debug assembler/C programs
- Analyze and describe simple logical circuits (Boolean expressions)
- Use Boolean algebrato reduce Boolean expressions

Competences
Having completed this course, students should be able to:

- Understand and explain the functionality of the components of basic computer architectures
- Use mathematical theory to understand low-level computer architecture and programming
- Construct small applications using low-level C and assembler programming
- Be able to integrate simple I/O devices in embedded applications

Topics

- Problem solving
- Number systems and number representations
- Digital electronics (gates, flip/flops, etc.)
- Boolean algebra
- CPU Architecture
- Memory
- Stack
- I/O
- Buses
- Instruction set
- Assembler programming
- Subroutines, calling conventions
- Introduction to timers and interrupts (AVR MCU)

- Logical operators
- Functions
- Logical operators
- Structuring source files for C
- Arrays and simple data structures
- Pointers in C
- Abstract data types in C
- UML in embedded Applications
- Atmel AVR
- Test Driven Development (TDD)

Teaching methods and study activities
Activities change between theory, self-study and exercises in the lab. The mandatory course tasks will typically be done in small project groups.

Each group must deposit 600 DKK for loan of necessary equipment.

Resources

- The AVR Microcintroller and Embedded System – using assembly and c; Muhammad Ali Macidi, Sermad Naimi, Sepehr Naimi. Prentice Hall, ISBN 0-13-800331-9 or 978-1292042565
- On-line resources
- Hand-outs

Evaluation
Internal examination.

Three-hour written exam (75%) and hand-in exercises (25%) with marks according to the 7-point grading scale.

Examination

Three-hour written examination.

Grading criteria
According to the 7-point grading scale.

Mark 12:
Awarded to students who have shown excellent comprehension of the above-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02
Awarded to students for the just acceptable level of comprehension of the required competences.

Additional information
Required workload for students is estimated to 275 hours where approximately 2/3 is self-study including exercises and examination preparation.

Responsible
Ib Havn

Valid from
1.8.2017

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;Business Information Systems;Cross Media;Embedded Engineering;

Offered by
ICT Engineering

IT-CON1

1.0

Computer Networking

ECTS
5

Prerequisites
General admittance requirements.

- SDJ1

Main purpose
The main purpose of the course is to gain a basic understanding of computer networks and the protocols used in the Internet.

Knowledge

- Knowledge about the nuts and bolts of the Internet
- Understand layered abstractions in TCP/IP
- Understand World Wide Web
- Understand the Internet's naming system
- Understand how electronic mail systems work
- Understand addressing in a TCP/IP network

Skills

- Configure a small office/home office (SOHO) network based on Ethernet technology
- Analyze network traffic using packet sniffer software
- Clients/server programming with sockets

Competences
Having completed this course, students should be able to:

- Configure computers to communicate on a network
- Understand and discuss protocols on a professional level

Topics

- TCP/IP
- Application level protocols
- IP addressing
- Network configuration
- Socket programming

Teaching methods and study activities
The required workload for students is estimated at 137 hours where approximately 2/3 is self study including exercises and examination preparation. Activities change between theory, self study and exercises.

Resources
James F. Kurose, Keith W. Ross. Computer Networking, A Top-Down Approach; Fifth Edition (International Edition). Pearson.

Evaluation
Internal examination.
25% of the grade will come from a test held during the course, 75% will come from the final examination.

Examination
Oral examination.

The duration of the oral examination is 20 minutes, including voting and marking.

Grading criteria
Students will be graded according to the Danish 7-point grading scale.

Mark 12:
Awarded to students who have shown excellent comprehension of the above-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02:
Awarded to students for a just acceptable level of comprehension of the required competences.

Additional information

Responsible
Erland Ketil Larsen

Valid from
1.8.2016

Course type
Global Business Engineering;Compulsory Course for GBE-ICT;5. semester;
ICT Engineering;Compulsory Course for all ICT Engineering;

Offered by
ICT
Engineering

CSE I1

10.0

Communication Skills for Engineers

ECTS
5

Prerequisites
General admittance requirements

Main purpose
The main purpose of the course is for students to develop the study skills required to study at a higher education programme. The course will strengthen the students’ skills in communicating successfully, including making project reports and giving oral presentations.

Knowledge

- State learning styles and personal profiles
- Understand the basic principles in intercultural communication
- Understand how to structure and deliver professional presentations
- Understand group development and group dynamics
- Understand the features of academic writing

Skills

Competences
Having completed this course, students should be able to:

- Analyse learning profiles and improve own learning outcome
- Communicate successfully in an intercultural context
- Demonstrate the knowledge of how to make groups successfull
- Prepare and give oral presentations
- Organise and structure reports
- Communicate correctly in accordance with basic grammar rules
- Communicate the use of an IT system using Use Cases

Topics
Study skills

- Learning styles
- Note-taking techniques
- Reading skills

Intercultural communication

- Relationships, social framework, time and power perception
- Communicating in a business perspective

Presentation skills

- Introduction to various presentation techniques
- Structuring presentations
- Being a presenter

Group work

- Group development, stages
- Group management
- Conflict management and prevention

Written skills

- Academic writing
- Reflective writing
- Cohesion
- Text structure

Teaching methods and study activities
12 weeks equalling 140 hours of work for the student. 3 compulsory written hand-in assignments. One test halfway through the semester. Class tutorials. Student presentations. Written and oral exercises.

Resources
Flensted and Petersen, “Box on Communicating”, Malling Beck, 2006.
Scott W. Ambler, “The elements of UML 2.0 Style” chapter 4.
Online resources

Evaluation
Students are assessed on the basis of class attendance - at least 80 % - and compulsory class activities.

Examination

Grading criteria
Approved/not approved

Additional information
Elements from the syllabus in the course Communication Skills for Engineers will be assessed at the exams in SEP I1 and SEP I2.

Responsible
Mona Wendel Andersen

Valid from
1.8.2014

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;1. semester;

Offered by
ICT Engineering

IT-DBS1

1.2

Database Systems

ECTS
5

Prerequisites
SDJ1 or similar

- IT-SDJ1

Main purpose
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).

Knowledge
Students will obtain theoretical knowledge on designing relational databases for practical application using a theoretical data modelling methodology. Students will obtain theoretical knowledge about relational algebra, UML notation for databases, E/R models, relational models, SQL, normalization, transaction handling and concurrency control.

Skills
Having completed this course, students will be able to:

- create ER-Models with UML
- understand and explain the relational model
- use Data Definition Language (DDL) to create databases
- use Data Manipulation Language (DML) to manipulate data in a database
- map ER-Models to Relational Models
- use basic SQL statements to create, replace, update and delete data in a database
- understand and use keys in relational databases
- understand and use joins
- handle the process of normalization to 3NF

Competences
Having completed this course, the students will be able to create database based applications using industry standard tools and methods.

Topics

Teaching methods and study activities
Lessons alternate between theory and practical exercises using the PostgreSQL relational DMBS. The course contains one or more compulsory assignments.

CATEGORY 1
Participation of lecturer and students
Initiated by the lecturer
55 hours- 40%

- Lessons, scheduled
- Exams and tests

CATEGORY 2
Participation of students
Initiated by the lecturer
50 hours - 37%

- Assignments, self-study
- Project and group work
- Homework and preparation for exams

CATEGORY 3
Participation of students
Initiated by students
25 hours - 19 %

- Homework and preparation for exams
- Self-study

- Study groups

CATEGORY 4
Participation of lecturer and students
initiated by students
5 hours - 4 %

- Project guidance

Resources
Connolly, Thomas and Begg, Carolyn: Database Systems (5th edition). Harlow, 2010, Pearson Education. ISBN: 987-0-321-52306-8

Evaluation
Permit criteria for attending examination

- Mandatory assignments handed in be-fore deadline and accepted.

Duration (grading included) app. 20 min/ 5 ECTS.

Examination

Oral Examination

Individual oral examination without preparation based upon course assignment(s).

Individual oral examination based upon a subject found by draw.

No preparation.

Allowed tools: All.

Internal examiner.

Grading criteria
Examinations account for 100 % of final grade.

Additional information

Responsible
Ole Ildsgaard Hougaard

Valid from
1.2.2018

Course type
Global Business Engineering;Compulsory Course for GBE-ICT;5. semester;
ICT Engineering;Compulsory Course for all ICT Engineering;2. semester;

Offered by
ICT Engineering

IT-DNP1

2.0

Internet Technologies, C# and .NET

ECTS
5

Prerequisites
IT-SDJ1, IT-DBS1, IT-RWD. Or similar courses. The course must be passed before graduation.

- IT-DBS1
- IT-RWD1
- IT-SDJ1

Main purpose
The purpose is to qualify the student to understand and implement the basic concepts of the C# programming language and the .NET developer platform.

Knowledge
The student should be able to understand:

- The fundamentals of .NET development and the common type system
- The C# programming language
- Web services
- Web applications
- Object-relational mapping
- Authentication and authorization
- Client-side vs. server-side programming

Skills
The student should achieve the skills:

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 web services
  - Navigate and use the managed .NET API
  - Understand security concerns of .NET applications
  - Apply unit testing and TDD principles to .NET applications
  - Use a command-line interface (CLI) toolchain
  - Deploy .NET applications

Competences
The student should be able to:

- Master the fundamentals of the C# programming language and the .NET developer platform
- Develop .NET applications and services as a part of a distributed system, herein account for communication protocols used

Topics

Teaching methods and study activities
The semester has 56 classroom lessons - four lessons once a week for 14 weeks. These lessons consist of exercises and teacher presentations. The total workload of the student is expected to be around 140 hours. Referring to the Study Activity Model, the workload is distributed as follows:

CATEGORY 1
Participation of lecturer and students

Initiated by the lecturer
42 hours - 30%

- Lessons, scheduled
- Exercises in class
- Project guidance
- Exam

CATEGORY 2
Participation of students
Initiated by the lecturer
42 hours - 30%

- Exercises
- Project and group work
- Homework

CATEGORY 3
Participation of students
Initiated by students
42 hours - 30 %

- Self-study
- Group work
- Literature search
- Preparation for exam

CATEGORY 4
Participation of lecturer and students
initiated by students
14 hours - 10 %

- Study guidance

Resources
Online material

Evaluation
Permit criteria for attending examination:

- Group reports including student’s name handed in before deadline.

Examination
Oral Examination.

The examination is a joint exam with IT-SEP3 and IT-SDJ3.

Group presentation followed by individual examination.

Group presentation of the IT-SEP3 project - 5 minutes per person.

Individual examination - 35 minutes including examination in IT-SEP3, IT-DNP1 and IT-SDJ3.
Allowed tools: All
Internal exam.

Grading criteria
IT-SEP3 accounts for 50% of final grade.
IT-DNP1 accounts for 25% of final grade.
IT-SDJ3 accounts for 25% of final grade.

Additional information

Responsible
Jakob Knop Rasmussen (JKNR)

Valid from
15.8.2018

Course type

ICT Engineering;Compulsory Course for all ICT Engineering;3. semester;

Offered by
ICT Engineering

IT-DSP1

1.1

Digital Signal Processing

ECTS
5

Prerequisites
Upper level mathematics equivalent to A-levels

- CAL I1

Main purpose
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.

Knowledge
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

Skills
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 analyze the frequency content of a signal

Competences
After successfully completing the course, the student will have acquired competences in:

- Explain sampling processes and how to determine the correct sampling frequency
- Describe signal processing applications
- Applying digital signal processing methods to analyze and interpret engineering problems
- Develop signal processing algorithms

Topics

- What is a signal?
- MATLAB
- Sampling theory
- A/D conversion
- Digital filters
- The frequency domain
- The Fast Fourier Transform and power spectrum
- Feature extraction (RMS, AUC, peak detection, peak latency, peak to peak, time intervals)

Teaching methods and study activities

Approximately 150 hours. The course is a mixture of lectures, hands-on MATLAB exercises, and hand-ins with approximately 1/3 of the time devoted to each part. Exercises are carried out in teams of 2-3 students.

Study Activity Model

Resources

Mark Owen - Practical Signal Processing © Cambridge University Press (ISBN 978-1-107-41182-1).
The MATLAB software will be integrated into this course.

Evaluation

The course is evaluated via an oral exam after course completion.

Grading will be done according to teh 7-scale, using an internal examiner.

Examination

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.

Grading criteria

According to the 7-point grading scale.

Internal examiner.

Mark 12:
Awarded to students who have shown excellent comprehension of the above-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02:
Awarded to students for the just acceptable level of comprehension of the required competences.

Additional information

For more information, please contact Line Lindhardt Egsgaard (lile@via.dk).

Responsible

Line Lindhardt Egsgaard

Valid from

1.1.2017

Course type

ICT Engineering;6. semester;7. semester;Elective for the specialization Business Information Systems;Elective for the specialization Cross Media;Elective for the specialization Enterprise Engineering;Elective for the specialization Web Engineering;Elective for the specialization Embedded Engineering;Electives;

Offered by
ICT Engineering

IT-ELE1

1.0

Basic Electronics (ICT) - 10 ECTS

ECTS
10

Prerequisites
General admittance requirements

Main purpose
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 the international report-writing standard (IMRAD).

Course secondary purpose:

- Introduction to measurement instruments in electronics
- Make systematic measurements and take precautions to minimize the effect of variance and error
- Analyse and visualize measured data
- Structured reports on experiments and be able to make correct conclusions

Knowledge
Having completed this course, students should have understanding of

- Statistics, Observation variance and error
- Ohm’s, Kirchhoff’s, Thevenin, Superposition and other laws used in electronics calculations
- Theory of basic analogue electronic components
- Cooling of electronic components (Heatzink, Compound)
- Filter properties
- Strain Gauges in Wheatstone Bridge
- Operation Amplifiers, Instrument Amplifier, transustor (NPN, PNP)

Skills
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
- Build and test prototype circuits
- Perform measurements on electronic circuits, using Digital Multi Meter and Pico-scope
- Solder components and wires

Competences
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

Topics
Electronics, Soldering, Instruments, Statistic, Report writing

Teaching methods and study activities
Lectures by teacher, group and individual work, tests, exercises, assignments, student presentations, class discussion conducted by teacher.

During the course the students will be able to design, simulate, build, test and make report on a simple electronic circuits in the ICT Laboratory.

Resources

Evaluation
Internal examination.
To qualify for the examination, the mandatory assignment/projects must have been carried out during the semester, within the set deadlines and must have been approved.
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.
The oral examination is based on electronics syllabus and three mandatory assignment reports:
It is a prerequisite that all of the three course assignment reports have been approved to attend the oral examination.
First two assignments should be performed in groups while the third will be done individually.

Examination
Oral examination. All written and electronic documents as approved aids.
In total the examination lasts for 20 minutes including marking.

Grading criteria
Grading:
Grading is according to the 7-point grading scale.

Mark 12:
Awarded to students who have shown excellent comprehension of the above-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02:
Awarded to students for the just acceptable level of comprehension of the required competences.

Additional information

Responsible
Poul Væggemose

Valid from
1.8.2016

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;4. semester;Embedded Engineering;

Offered by
ICT Engineering

IT-ESB1

2.0

Engineering Science and Business

ECTS
5

Prerequisites
The course requires at least 1 year of studies at an IT-related programme at AP degree level and up.

Main purpose
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 main purpose of this 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:

- Clarify which consequences their scientific point of departure has for their choice of methods, and thus what this means for analysis and assessment of practice as well as to identify basic theory of science problems related to engineering science
- Account for the distinctive character of IT and its relation to other academic disciplines
- Account for what it means to be an ICT 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 functions.
- Finally, the students will come to understand and master the knowledge-based challenges that one meets in a modern information society.

Knowledge
After successfully completing the course, the student will have gained knowledge about:

Science:

What the profession of ICT Engineering is
What knowledge and science are
What role knowledge plays in ICT 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
Business:

Basic concepts of business economics in IT companies
Understanding the basic structure and contents of annual reports
Knowledge about budgeting and calculations in IT companies
Knowledge about capital budgeting.

Skills
After successfully completing the course, the student will be able to:

Science:

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
Business:
Discuss and understand key elements of the economic side of IT and business

Competences
After successfully completing the course, the student will have acquired competences in:

- Using scientific methods to solve practical engineering problems
- Using Annual Reports as a source for getting information about companies
- Understanding of aim and techniques for budgeting and calculations

Topics

Teaching methods and study activities

The mode of instruction will be classroom based and will involve short lectures by the teacher, classroom discussions, group work and student based group presentations. The students are also expected to read the literature before classes. The total workload for the student is expected be around 150 hours.

Referring to the Study Activity Model, the workload is distributed at follows:

CATEGORY 1: 30%
Participation of lecturer and students
Initiated by the lecturer
Lessons, scheduled
Project guidance

CATEGORY 2: 50%
Participation of students - Initiated by the lecturer
Assignments, self-study
Project and group work
Homework and preparation for presentations
Evaluation of the teaching

CATEGORY 3: 20 %
Participation of students - Initiated by students
Homework and preparation for presentations
Self-study
Project work
Study groups
Literature search

Resources

Science: A compilation of texts will be made available online
Business: Literature will be available in StudyNet.

Evaluation

Students are assessed

1) On the basis of class attendance - at least 80 % in both Business and Science;

2) On compulsory activities in Science

3) One compulsory activity in Business

If a student fails to meet one or more of the above requirements for passing the course, the student will have to do a number of written works the following semester and present this for the teachers and the other students that failed the course.

Examination

There is no examination as the course is evaluated based on attendance and compulsory assignments.

Grading criteria

Additional information

Responsible

Lars Bech Sørensen (LBS)

Valid from

1.8.2018

Course type

ICT Engineering;Compulsory Course for all ICT Engineering;4. semester;Business Information Systems;Cross Media;Embedded Engineering;
Software Ingeniør;Compulsory for Software Ingeniør;

Offered by
ICT Engineering

IT-HWP1

1.0

Hardware Oriented programming

ECTS
5

Prerequisites

- IT-CAL1
- IT-SDJ1

Main purpose
The purpose of the course is:

- To provide the student 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

Knowledge

Skills

Competences
Having completed this course, students should be able to:

- have knowledge to read datasheets for electronics components
- understand the architecture of micro-controllers
- master IO-Ports
- master Interrupts
- master Timers
- master Analog input/output
- understand how to divide the software into logical abstraction layers
- be able to implement software for micro-controllers in C
- understand how to design and implement simple device drivers
- to exemplify the above topics in small applications.

Topics

- Architecture of industrial micro-controllers.
- Memory and IO-systems.
- Basic peripherals.
- Interfacing to the analogue world.
- Interrupts and exceptions.
- Developing software for embedded systems.
- Programming device drivers.
- Unit testing of embedded C

Teaching methods and study activities

Required workload for students is estimated to 137 hours. ~2/3 of the workload is self-study by the student.
Activities change between theory, tasks, programming exercises and mini projects.

During the semester 4-5 compulsory assignments will be given. The result of these assignments will lead to 25% of the final assessment grade for the student.

Each group must deposit 600 DKK for loan of necessary equipment.

Resources

Muhammad Ali Mazidi, Sarmad Naimi & Sepehr Naimi: The AVR microcontroller and embedded system.

Copies from various books and notes.

Evaluation

Internal examination.
Final grade based 25% on tuition activities selected as compulsory by the teacher carried out within the set deadlines and approved, and 75% from the oral examination.

Examination

The exam is oral and it takes 20 minutes per student. The exam is in two parts. First part is a presentation and questions related to the solution to an assignment made in the course. Second part is a drawn question related to the course subjects.

Grading criteria

Mark 12:
Awarded to students who have shown excellent comprehension of the above-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02:
Awarded to students for a just acceptable level of comprehension of the required competences. According to the 7-point grading scale. comprehension of the required competences.

Additional information

Responsible

Ib Havn

Valid from

1.2.2015

Course type

ICT Engineering;Compulsory Course for all ICT Engineering;7. semester;Compulsory for the specialization Embedded Engineering;Electives;

Offered by
ICT Engineering

IT-INO1

2.0

Engineering Innovation (ICT)

ECTS
5

Prerequisites

Main purpose

Knowledge
After having successfully completed the course, the students will have gained:

An understanding of their own professional identity in and of itself, as well as contrasted and compared to other fields of engineering
An understanding of innovation and its uses within the field of engineering
Knowledge about innovative processes within the field of engineering

Skills
After having successfully completed the course, the students will be able to:

Engage in innovative processes in a Cross-/inter-/multidisciplinary setting in order to conceive, plan and execute their ideas
Work methodically with innovation
Apply relevant models to the implementation of product and concept development

Competences
After having successfully completed the course, the students will have gained competences in:

Introducing innovative ideas into project work
Assess when innovation is needed and what the value of initiating an innovative process will be
Contributing own professional skills in teams with the objective of solving problems by using innovative processes and models

Topics
Clarifying multidisciplinary group competencies
The history of the engineer and engineering
VIA engineering in an innovative perspective
Field research
Field trip
The 4D model: Discover, iDiate, Design and Deliver
Process reflection
Innovation competition

Teaching methods and study activities
Engineering Innovation is a three-week comprehensive course in which the students work partly in their own faculty (one week, see separate description) and partly cross-faculty in VIA Engineering (two weeks):

Week 49-50: Working in multidisciplinary groups in VIA Engineering (Monday December 4 – Friday December 15, 2017) .Throughout the course, the students will work in groups gaining innovative tools and using these for solving specific challenges posed by actual companies. Additional information about the content and scope of the group challenges and deliveries will be announced during week 49.

External partners from the companies who posed the challenges assess the students. The students are assessed in terms of their written deliveries and their oral presentation, emphasis being on their idea, the process described for working with the idea, and the final product and/or solution.

Resources

Evaluation
In order to qualify for an approval, the students: Must have an attendance of 100% at the Engineering Innovation during the three-week period.

Examination

Group presentation of project Friday December 15.

Grading criteria
Full participation in all activities during the course (check-in/out each day). Approved/not-approved.

Additional information

Responsible
Behnam Boujarzadeh

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;Business Information Systems;Cross Media;Embedded Engineering;

Offered by
ICT Engineering

IT-INP1

2.0

Engineering Internship (ICT)

ECTS
30

Prerequisites
To start an internship the student must have passed all courses, projects and workshops of the previous semesters. An application for an exemption must be submitted in writing to the study counsellor of the programme.

Main purpose

Knowledge
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

Skills
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

Competences
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.

Topics
The internship aims to give students an introduction to work as an engineer.

Teaching methods and study activities
The internship consists of minimum 20 weeks of full time work corresponding to approx. 740 hours of work.
Furthermore, documentation, participation in workshop for coming interns etc. account for approx. 85 hours of work.

Study Activity Model:

CATEGORY 1 Participation of lecturer and students, initiated by the lecturer 5 hours/0.5%
CATEGORY 2 Participation of students, initiated by the lecturer: 75 hours/9%
CATEGORY 3 Participation of students, initiated by students: 740 hours/90 %
CATEGORY 4 Participation of lecturer and students, initiated by students: 5 hours/0.5%

Resources

Evaluation
In order to get an internship evaluated, the student must fulfil the following requirements concerning mandatory assignments:

- Expected outcome/specific learning targets for the internship position
- Company presentation
- Logbook
- Main academic assignment(s)

- Final reflections
- Recommendation letter from the company
- Participation in workshop for coming interns

Grading criteria
The course is evaluated by Approved/Not approved.

Additional information
Deadline for signing of an Internship agreement:

Spring internship: 15th December
Autumn internship: 1st July

Responsible
Karin Larsen

Valid from
1.8.2017

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;5. semester;

Titel	Lego Robot Lab 1
Kode	IT-LRL1
Ver	1.0
Sprog	EN
Udbyder	ICT Engineering
Kursusansvarlig	Troels Mortensen
ECTS-point	5
Forudsætninger	General Admission Requirements
Interne forudsætninger	
Formål	Enabling the student to understand the basic concepts and techniques of embedded system development using Java to program a Lego Mindstorms robot.
Nøgleord	
Indhold	• Real world interaction • Sensor input and responds to stimuli: light, pressure, ultrasound, sound, sampling, accuracy • Output: motor control, sound, LCD display • Searching and sorting
Læringsmål	
Viden	• Describing the elements of a Java program • Understanding motors and sensors • Understand how programs interact with the real world
Færdigheder	After having completed this course, the student should be able to: • Write basic Java programs • Use the Java programming language to control a robot • Program a robot to react to its environment using Java • Implement and use different searching and sorting algorithms
Kompetencer	
Undervisningsform og aktiviteter	Estimated workload for students is 137 hours. Activities will vary between theory presented on class and practical exercises in the laboratory. The students will build a robot and complete a programming project towards the end of the semester. Each group must deposit 600 DKK for loan of necessary equipment.
Evaluering	To qualify for the examination, the student must show an attendance of at least 75% of the lessons during the semester.

25% of the grade will come from a test held during the course, 75% will come from the final examination.

Internal examination.

Eksamen

Oral exam based on the final hand-in. At the end of the semester, the students will hand-in an assignment. The students will present the assignment in the form of a demonstration, followed by questions about the programming and the logic behind the robot.

The duration of the oral examination is 20 minutes, including voting and marking.

To attend the exam it is a condition that course activities selected as compulsory by the teacher have been carried out within the set deadlines and approved.

Karakterbeskrivelse

According to the 7-point grading scale.

Mark 12:
Awarded to students who have shown excellent comprehension of the above-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02
Awarded to students for the just acceptable level of comprehension of the required competences.

Studieaktivitetsmodel

CDIO

Ressourcer

Tutorials from the following homepage:
http://lejos.sourceforge.net/nxt/nxj/tutorial/index.htm and hand-outs.

All students must be a part of a group and all groups must pay a deposit for the LEGO NXT equipment.

Yderligere oplysninger

The LEGO NXT equipment must be handed in at a date set by the teachers.

Gældende fra

01-08-2016

Godkendt af

BY

BY udbud

CE

CE udbud

CE Exchange

CE Exchange udbud

FI

FI udbud

GBE

Compulsory Course for GBE-ICT; 4. semester; Elective for the specialization Information Technology and Management

GBE udbud

2018 Feb-Jun

GBE Exchange

GBE Exchange udbud	
ICT	
ICT udbud	
ICT Exchange	
ICT Exchange udbud	
MA	
MA udbud	
ME	
ME udbud	
ME Exchange	
ME Exchange udbud	
SE	
SE udbud	
Arkiv	Nej
SortCourses	
SortCourses_GBE	
Efteruddannelse	Nej
Efteruddannelse_tid	
Efteruddannelse_pris	
SortCourses_ICT	
SortCourses_CE	

Version: 11.0
Oprettet af kl. 21-08-2016 15:21 af Lis Therkildsen (LITH) | VIA
Senest ændret kl. 13-01-2018 17:08 af Lis Therkildsen (LITH) | VIA

Offered by
ICT Engineering

IT-PCL1

1.0

Programming Concepts and Languages

ECTS
5

Prerequisites
Considerable programming experience with e.g. Java, C# and C.

Main purpose
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

Knowledge

Skills

Competences

Topics
Overview of programming concepts, paradigms and languages
Imperative paradigm with Imperative Programming in Python
Object-Oriented paradigm with Object-Oriented Programming in Scala
Concurrent paradigm with Concurrent Programming in Erlang
Functional paradigm with Functional Programming in F#
Integration of different programming paradigms in system development.

Teaching methods and study activities
Activities will vary between theory, programming exercises, and mini project work.

Study Activity Model

Resources
Some papers about the general programming concepts and languages.
A book about functional programming in F#.
Lesson notes.

Evaluation
Written programming hand-in exercises and presentations: 30 %.

Examination

Three-hour written examination (70 %) and hand-in exercises/presentations (30 %) with marks according to the 7-point grading scale.

External examination.

Grading criteria
According to the 7-point grading scale.

Mark 12 Awarded to students who have shown excellent comprehension of the below-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02 Awarded to students for the just acceptable level of comprehension of the required competences.

Additional information
Required workload for students is estimated to 150 hours.

Responsible
Joseph Okika

Valid from
1.8.2017

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;6. semester;Electives;

Offered by
ICT Engineering

IT-PME1

1.0

Process Management for ICT Engineering

ECTS
5

Prerequisites
Project experience from semester projects and internship.

Main purpose
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.

Knowledge
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.

Skills
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.

Competences
To complete this course the students must make hand-in:

- "Test plan” document for a project
- “Test Specification” document for a project
- “Project relations to CMMI model” document for a project.

Topics
Software Process Improvement,
Software Development Strategy,
Software Testing Principles,
Improve Component Testing,
How to Break Software.

Teaching methods and study activities
Activities alternate between theory, self-study, group work, exercises and hands-on experiments.

Additional information:

Required workload for students is estimated at 100 hours where approximately two thirds are self-study, including exercises and preparing for the examination.

Description:
Capability Maturity Model Integration (CMMI) is a process improvement approach that helps organizations improves their performance. CMMI can be used to guide process improvement across a project, a division, or an entire organization.
CMMI in software engineering is a trademarked process improvement approach that provides organizations with the essential elements for effective process improvement.
According to the Software Engineering Institute (SEI, 2008), CMMI helps "integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes."

Resources

1.
Practical Insight into CMMI® (2nd Edition) ebrary Reader
Author: Kasse, Tim
Publisher: Artech House
Released: 2008
Subjects: Capability maturity model (Computer software) Computer software Development.

2.
How to Break Software, A Practical Guide to Testing,
Author: Whittaker James
Released: 2002
ISBN: 0-201-79619-8

3.
How to Break Software, Security
Author: Whittaker James
Released: 2003
ISBN: 0-321-19433-0

Evaluation
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.

Examination

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.

Grading criteria
Grading is according to the 7-point grading scale.

Mark 12:
Awarded to students who have shown excellent comprehension of the above-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02:
Awarded to students for the just acceptable level of comprehension of the required competences.

Additional information

Responsible
Poul Væggemose

Valid from
1.2.2012

Course type
Global Business Engineering
ICT Engineering;6. semester;Elective for the specialization Enterprise Engineering;Elective for the specialization Embedded Engineering;Electives;

Offered by
ICT Engineering

IT-RIE1

1.2

Real-Time Programming, Interfacing and Electronics

ECTS
10

Prerequisites
Similar courses to AJP1 (especially thread programming) at your home institution. Similar courses to CAL1 (Computer Architecture and C-Programming) at your home institution.

- IT-AJP1
- IT-CAL1

Main purpose
The main purpose of the course is to provide students with the qualifications needed to understand central concepts and characteristics about real-time programming, and to give the students knowledge about electronics used to interface micro-controllers to a number of sensors and actuators.

Knowledge
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.
- Basic understanding of datasheets for electronics components
- Explain and understand simple electronics interfaces typically used in embedded systems
- Understand low-level protocols (Datalink- and Physical Layers).

Skills
Having completed this course, the student should be able to:

- Write functioning real-time programs in C using FreeRTOS
- Analyze a simple real-time design for schedulability, deadlocks, utilization etc.
- Design and calculate components for simple electronics interfaces
- Design and implement low-level protocols.

Competences
Having completed this course, students should be able to:

- master and use simple real-time operating systems
- be able to analyze/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
- be able to design simple electronic hardware interfaces to sensors and actuators
- have knowledge to read datasheets for electronics components
- understand low-level protocols, CRC etc.

Topics
Real-time Programming:

- C Programming
- Introduction to FreeRTOS
- Real-time concepts

- Tasks
- Scheduling – RMS, DMPO, Arbitrary Deadlines, EDF
- Queues
- Timers
- Clocks
- Memory management
- Resource sharing
- Semaphores and mutexes
- Synchronization
- Priority Inversion
- Priority - static, dynamic
- Low-level protocols
- CRC
- Hamming-codes

Interfacing and Electronics:

- Introduction to understand datasheets for electronics components
- Relay drivers
- H-Bridge and Pulse Width Modulation (PWM)
- Analogue to digital and visa versa converters
- Printed Circuit board design – schematics and PCB-layout

Test:

- Unit test in C
- Simple integration test

Teaching methods and study activities
Activities change between theory, tasks, practical exercises and lab-work.

Each group must deposit 600 DKK for loan of necessary equipment.

Resources
Paul Scherz, “Practical Electronics for Inventors”, 2nd edition McGraw-Hill, 2007, ISBN: 0-07-145281-8

Alan Burns and Andy Wellings, “Real-Time Systems and Programming Languages”, 2009, ISBN 978-0-321-41745-9 Additional notes and on-line books

Evaluation
Internal examination.

Examination

Four hour written exam.

All non-electronic aid allowed (except simple calculators that cannot communicate).

Grading criteria
According to the 7-point grading scale.

Mark 12
Awarded to students who have shown excellent comprehension of the below-mentioned competences. A few minor errors and shortfalls are acceptable.

Mark 02
Awarded to students for the just acceptable level of comprehension of the required competences.

Additional information
Expected workload 11 weeks of 10 lessons per week.

Required workload for students is estimated to 275 hours where approximately 2/3 is self-study including exercises and examination preparation.

Responsible
Lars Bech Sørensen

Valid from
1.2.2018

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;6. semester;Embedded Engineering;Electives;

Offered by
ICT Engineering

IT-RWD1

2.0

Responsive Web Design

ECTS
5

Prerequisites
General admittance requirements.

Main purpose
The main purpose of the course is to introduce a set of theories, tools, and practicable methods in order for students to obtain a profound knowledge and practical skills of web development with the ability of creating web sites running on PC and mobile devices using basic web programming and design combined with jQuery.

Knowledge
The student will gain knowledge of the following topics:

- HTML5
- CSS3
- W3C Validation
- Responsive web-Design (Multiplatform)
- jQuery
- Bootstrap

Skills
Having completed this course, students will be able to:

- Create web sites using Hyper Text Markup Language (HTML5)
- Style web pages using Cascading Style Sheets (CSS3)
- Using responsive web-Design to design web sites that are platform independent
- Create “smart” web sites using jQuery, and Bootstrap

Competences
To obtain the fundamental knowledge needed to:

- Create platform independent web applications

Topics

Teaching methods and study activities
Required workload for students is estimated at 150 hours where approximately two thirds are self-study, including exercises and preparation for the examination. Activities alternate between theory, self-study, and exercises.

CATEGORY 1
Participation of lecturer and students
Initiated by the lecturer
55 hours - 37%

- Lessons, scheduled
- Excursions
- Project guidance
- Laboratory work
- Exams and tests

CATEGORY 2
Participation of students
Initiated by the lecturer
30 hours - 20%

- Assignments, self-study
- Project and group work
- Homework and preparation for exams
- Evaluation of the teaching

CATEGORY 3
Participation of students
Initiated by students
55 hours - 37%

- Homework and preparation for exams
- Self-study
- Project work
- Study groups
- Literature search

CATEGORY 4
Participation of lecturer and students
Initiated by students
10 hours - 6%

- Debate meetings
- Study guidance

Resources

[Duckett, 2011] : HTML & CSS Design and build Webstites, Jon Duckett, First Edition, ISBN: 978-1-118-00818-8

[LaGrone, 2013] : HTML5 and CSS3 Responsive Web Design Cookbook, ISBN: 978-1-84969-544-2

[Duckett, 2014] : Javascript & jQuery, Jon Duckett, First Edition, ISBN: 978-1-118-53164-8

https://www.w3schools.com/
http://getbootstrap.com/
http://jquery.com/

Evaluation

Permit criteria for attending examination:
* Mandatory course activities completed
* Mandatory assignments handed in before deadline
and accepted.
* Attendance (75%)
The student must have an attendance of at least 75% in order to qualify for the exam. If this requirement is not met, the student will automatically fail the ordinary exam.

Examination

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

Grading criteria

For ordinary examinations:

Examinations account for 100 % of final grade.

For re-examinations:

Examination will count for 100 % of final grade.

Additional information

Responsible

Line Lindhardt Egsgaard (LILE)

Valid from
1.2.2018

Course type
Global Business Engineering;Compulsory Course for GBE-ICT;4. semester;
ICT Engineering;Compulsory Course for all ICT Engineering;1. semester;
Software Ingeniør;Compulsory for Software Ingeniør;1. semester;

Offered by
ICT Engineering

IT-SDJ1

2.0

Software Development with UML and Java

ECTS
10

Prerequisites
General admittance requirements. The course must be passed before graduation.

Main purpose
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.

Knowledge
The student should be able to understand:

- Java programming constructs
- Basic types in Java and their applicability
- The role of fields, methods, constructors, and references in Java programming
- The relationship between array, array length, indices and elements in Java
- The elements of UML class diagrams, activity diagrams and sequence diagrams.

Skills
The student should achieve the skills:

- Create and use objects in Java
- Implement programs in Java with
  o Fields, constructors and methods
  o Inheritance
  o Association, aggregation and composition
  o Arrays and ArrayLists
  o File IO persistence
  o Exceptions
  o Simple GUI's
- Use best practices for writing and documenting Java source code

Competences
The student should be able to:

- Master and use the basic object-oriented concepts, including relationships, collaboration and poly-morphism
- Implement smaller programs in Java including simple GUIs
- Implement smaller systems from a UML design

Topics

Teaching methods and study activities
The semester has 110 classroom lessons. A number of tests and assignments will be given during the course. One of these tests represent 25% of the exam, see Evaluation below.

The course is held concurrently with Workshop in Basic Programming I (WS1).

CATEGORY 1
Participation of lecturer and students
Initiated by the lecturer
84 hours - 30%

- Lessons

- Exercises in class
- Exam

CATEGORY 2
Participation of students
Initiated by the lecturer
112 hours - 40%

- Exercise, assignments and hand-ins
- Group work
- Homework

CATEGORY 3
Participation of students
Initiated by students
56 hours - 20%

- Preparation for exam
- Self-study
- Group work
- Literature search

CATEGORY 4
Participation of lecturer and students
Initiated by students
28 hours - 10%

- Study guidance
- Study group meetings

NOTE: For GBE students there is only 8 lessons per week, as opposed to ICT, where there are 10 lessons per week.

NOTE: Compulsory attendance does not apply for GBE students.

Resources

Tony Gaddis, "Starting Out with Java - early objects", 5th edition Addison-Wesley, 2015

Evaluation

Permit criteria for attending examination:

Mandatory course activities completed.

The student must have an attendance of at least 75% in order to qualify for the exam. Students who do not have at least 75% attendance will automatically fail the ordinary exam.

Examination

* 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 75% of the final grade while the remaining 25% comes from an oral test conducted in the middle of the course.

* Allowed tools: All

* External examiner.

Grading criteria

* Examinations account for 75% of final grade.

* A test account for 25% of final grade.

* Rexams account for 100 % of the grade.

Additional information

Responsible
Steffen Vissing Andersen (SVA)

Valid from
1.8.2017

Course type
Global Business Engineering;Compulsory Course for GBE-ICT;3. semester;Elective for the specialization Information Technology and Management;
ICT Engineering;Compulsory Course for all ICT Engineering;Project;1. semester;
Software Ingeniør;Compulsory for Software Ingeniør;1. semester;

Offered by
ICT Engineering

IT-SDJ2

2.1

Software Development with UML and Java 2

ECTS
10

Prerequisites
Completed the 1st semester ICT Engineering course “Software Development with UML and Java 1” or a similar course. The course must be passed before graduation.

- IT-SDJ1

Main purpose
The purpose is to qualify the student to understand and master the concepts and techniques of object-ori-ented 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

Knowledge
The student should be able to understand:

- System architecture
- Different methods for testing
- Concurrency
- System deployment
- Design patterns
- Client/server structure

Skills
The student should achieve the skills:

- Implement design patterns in Java
- Test software using different testing techniques, including (but not limited to) JUnit testing, System testing, etc.
- Create jar files
- Implement thread-safe classes and multi-threaded programs
- Make programs communicate using client-server technologies

Competences
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

Topics

Teaching methods and study activities
The semester has 110 classroom lessons. These lessons will consist of teacher presentations and exercises.

The course will include one or more course assignment(s).

The course is held concurrently with Workshop in Basic Programming 2 (WS2).

CATEGORY 1
Participation of lecturer and students
Initiated by the lecturer
56 hours - 20%

- Lessons
- Exercises in class
- Exam

CATEGORY 2
Participation of students
Initiated by the lecturer
98 hours - 35%

- Exercise, assignments and hand-ins
- Project and group work
- Homework

CATEGORY 3
Participation of students
Initiated by students
70 hours - 25%

- Preparation for exam
- Self-study
- Group work
- Literature search

CATEGORY 4
Participation of lecturer and students
Initiated by students
56 hours - 20%

- Study guidance
- Study group meetings

Resources

Evaluation

Permit criteria's for attending examination:

Mandatory course activities completed

- A number of presentations

Course assignment handed in before deadline

Examination

- Individual oral examination without preparation based upon course work.
- The student will draw from a pool of previously known questions.
- The student will explain concepts and theories from the course, using the course work as reference.
- The student will start with a prepared presentation.
- External examiner.
- The course must be passed before graduation.

Grading criteria

Examinations account for 100% of final grade.

Additional information

Responsible

Troels Mortensen

Valid from

1.8.2018

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;2. semester;

Offered by
ICT Engineering

IT-SDJ3

2.0

Software Development of Distributed Systems

ECTS
5

Prerequisites
Condition for following IT-SDJ3 is that you have followed the courses IT-SDJ1 and IT-SDJ2, or similar courses at your home institution. ● IT-SDJ1 ● IT-SDJ2 The course must be passed before graduation.

- IT-SDJ1
- IT-SDJ2

Main purpose
The students should be introduced to basic theory of distributed systems and be able to design and implement a distributed system.

Knowledge
The students will

- be introduced to various distributed system types (e.g. client/server, peer-to-peer)
- get knowledge of the 3-tier architecture
- get knowledge of various distributed communication methods
- be introduced to examples of distributed algorithms

Skills
The students will be able to:

- use various middleware (e.g. Web Services)
- choose middleware for a given distributed system

Competences
The students will be able to

- design the architecture of a distributed system using the 3-tier model
- design and implement a distributed and on different platforms using various middleware

Topics

- client/server systems
- peer-to-peer systems
- 3-tier architecture
- web services middleware
- object orientation in distributed system
- concrete middleware systems
- distributed synchronization and distributed transactions

Teaching methods and study activities
The semester has 56 classroom lessons. These lessons consist of exercises and teacher presentations. The total workload of the student is expected to be around 140 hours.

Referring to the Study Activity Model, the workload is distributed as follows:

CATEGORY 1
Participation of lecturer and students
Initiated by the lecturer
42 hours - 30%

- Lessons, scheduled
- Exercises in class
- Project guidance
- Exam

CATEGORY 2
Participation of students
Initiated by the lecturer
42 hours - 30%

- Exercises
- Project and group work
- Homework

CATEGORY 3
Participation of students
Initiated by students
42 hours - 30 %

- Self-study
- Group work
- Literature search
- Preparation for exam

CATEGORY 4
Participation of lecturer and students
initiated by students
14 hours - 10 %

- Study guidance

Resources
Coulouris & Dollimore & Kindberg & Blair, Distributed Systems -- Concepts and Design, Fifth Edition, Addison-Wesley, 2012.
Notes

Evaluation
Permit criteria for attending examination:

- Group reports including student’s name handed in before deadline.

Examination
Oral Examination

The examination is a joint exam with IT-SEP3 and IT-DNP1.

Group presentation followed by individual examination.

Group presentation of the IT-SEP3 project - 5 minutes per person.

Individual examination - 35 minutes including examination in IT-SEP3, IT-SDJ3 and IT-DNP1.

Allowed tools: All

Internal exam.

Grading criteria
IT-SEP3 accounts for 50% of final grade.
IT-SDJ3 accounts for 25% of final grade.
IT-DNP1 accounts for 25% of final grade.

Additional information

Responsible

Ole Ildsgaard Hougaard (OIH)

Valid from
1.8.2018

Course type
ICT Engineering;Compulsory Course for all ICT Engineering;3. semester;

Offered by
ICT Engineering

IT-SEP1

2.0

Semester Project: Single User System

ECTS
5

Prerequisites
General admittance requirements.

Main purpose
The purpose is to develop and document a single user system as well as demonstrate an acquisition of process skills.

Knowledge
The student should be able to understand:

- The Waterfall method as a software development process
- Requirements, use case modelling and activity diagrams in relation to the analysis stage
- Class diagram and sequence diagrams in relation to the design stage
- Basic use case testing
- Group roles

Skills
The student should achieve the skills to:

- Design and describe an object oriented model using UML
- Practice implementing smaller software systems in Java
- Create a website
- Present considerable skills for presentation, both written and oral
- Present a project report in a well-structured manner
- Describe a project in a process report
- Solve a specific task in collaboration with group members

Competences
The student should be able to:

- Analyse, design, implement and document a single user system from a case story
- Create small software systems with UML according to the object-oriented method
- Control and structure a project as it progresses
- Reflect on the group and individual leaning processes

Topics

Teaching methods and study activities
Lessons, input, meetings, project work, facilitation.
CATEGORY 1
Participation of lecturer and students
Initiated by the lecturer
14 hours -10%

- Lessons
- Project facilitation, meetings with supervisors
- Exam

CATEGORY 2
Participation of students
Initiated by the lecturer
28 hours -20 %

- Hand ins
- Project work
- Homework

CATEGORY 3
Participation of students
Initiated by students
70 hours - 50 %

- Preparation for exam
- Self-study
- Project work
- Literature search

CATEGORY 4
Participation of lecturer and students
Initiated by students
28 hours -20 %

- Supervisor meetings

Resources

Online resources.

Evaluation

Qualifying criteria for attending examination:

Group reports including students´ name handed in before deadline.

Examination

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.

Grading criteria

100% of the grade is based on a total assessment of the written materials, oral presentation and individual performance at the examination.

Additional information

Responsible

Steffen Vissing Andersen (SVA)

Valid from

15.8.2017

Course type

ICT Engineering;Compulsory Course for all ICT Engineering;Project;1. semester;
Software Ingeniør;Compulsory for Software Ingeniør;1. semester;