Curriculum Autumn 2017*

Bachelor in IT – Intelligent Systems
Faculty of Technology

All information revised as of October 2016. Note that all curricula may be updated*
1 Introduction

Intelligent systems automatis work tasks and create intelligent environments; they make machines talk to each other — for instance in mobile payment, health, traffic, security or surveillance. In this study programme you will learn to map needs, and to make and implement solutions for consumer technology such as beacons, mobile phones and smart cities and homes. A thorough introduction is given to machine-to-machine technology, automatisation and control, programming, networks, the Internet of Things, and sensor networks.

To achieve this qualification you need knowledge of design processes, sensors, tools, and technologies. Intelligent systems provide you with the tools and knowledge you need, including a thorough introduction to sensors, networks, design, and implementation. The study prepares you to take the digital life of the future one step further. And you will learn to plan, design and implement projects of smart systems and the Internet of Things.

1.1 Curriculum overview

The course structure is shown in the matrix below.

<table>
<thead>
<tr>
<th>Sem6</th>
<th>Research Methods (7.5 ECTS)</th>
<th>Bachelor's Thesis (22.5 ECTS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sem5</td>
<td></td>
<td>Data Science (7.5 ECTS)</td>
</tr>
<tr>
<td></td>
<td>Elective (7.5 ECTS)</td>
<td>Machine-to-Machine Communication (7.5 ECTS)</td>
</tr>
<tr>
<td></td>
<td>Project Software Engineering (15 ECTS)</td>
<td>Machine Learning (7.5 ECTS)</td>
</tr>
<tr>
<td></td>
<td>Mobile Development (7.5 ECTS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Algorithms and Data Structures (7.5 sp)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Programming in C for Linux (7.5 sp)</td>
<td></td>
</tr>
<tr>
<td>Sem4</td>
<td></td>
<td>C++ Programming (7.5 ECTS)</td>
</tr>
<tr>
<td></td>
<td>Object Oriented Programming 2 (7.5 ECTS)</td>
<td>Embedded Systems (7.5 ECTS)</td>
</tr>
<tr>
<td></td>
<td>Information Security (7.5 ECTS)</td>
<td></td>
</tr>
<tr>
<td>Sem3</td>
<td>Elective (7.5 ECTS)</td>
<td>Iterative Web Project (7.5 ECTS)</td>
</tr>
<tr>
<td></td>
<td>Databases 1 (7.5 ECTS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Object Oriented Programming 1 (7.5 ECTS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Digital Technology (7.5 ECTS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creative Web Project (7.5 ECTS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.5 ECTS credits</td>
<td>7.5 ECTS credits</td>
</tr>
<tr>
<td></td>
<td>7.5 ECTS credits</td>
<td>7.5 ECTS credits</td>
</tr>
</tbody>
</table>
Yellow boxes denote that the course is taught jointly with one or more other study programmes. Green boxes show the location of optional courses (electives) in the curriculum. In the 2nd semester students choose an elective from a pool of optional courses across faculties and programmes, in addition to the electives offered by the Faculty of Technology.

**Electives in the 2nd semester (Spring)**
- DS2100 Animation (7.5 ECTS)
- DS2200 Digital Culture (7.5 ECTS)
- PG2201 Unity Development (7.5 ECTS)

**Electives in the 5th semester (Autumn), Bachelor in Intelligent Systems**
- PG3300 Software Design (7.5 ECTS)
- PG5600 iOS Programming (7.5 ECTS)
- PRO300 Virtual Reality Project (7.5 ECTS)

Bachelor in Intelligent Systems is a programme offered by the Faculty of Technology at Westerdals Oslo ACT. The study lasts 3 years and awards the degree Bachelor in IT. The first year is a joint study for all the IT programmes and gives basic qualifications in programming, project work, system development, security, data technology and databases.

In the second year the emphasis is on intelligent systems and sensors. Further, an introduction to the ecosystem is given as well as to interface design, advanced Java, and networks. Project software engineering is taught across both semesters, gathering the knowledge gained and using it in a fairly extensive group-based project.

The third year of study has its focus on the total knowledge and understanding of intelligent systems and architecture, by continuing the in-depth study of the platform and by providing good knowledge of development, sensors and design. Learning about autonomous systems, focusing on automation, artificial intelligence and future visions, provides an understanding of the totality of these IT solutions. The third year also includes joint courses that are important in the bachelor’s programmes in IT: an introduction to research methods including qualitative as well as quantitative methods and an introduction to entrepreneurship, and the bachelor’s thesis.

1.2 About the programme

Bachelor in IT specialising in Intelligent Systems aims to educate candidates with the qualifications to map the needs, create and implement solutions for consumer technology such as beacons, mobile phones and smart homes and cities. The closeness to the industry, cooperating in practical work in cross-disciplinary teams gives students essential experiences in working with complex issues, as well as a broad basis for success in their future careers in intelligent systems; systems that automate work tasks and create intelligent environments that make the machines talk to each other in e.g. mobile payment, health, traffic, security or surveillance.
The programme has a jointly taught first year for all the specialisations leading to the Bachelor in IT, followed by two years focusing on the specialisation and on practical project work, often performing work for external employers. Out of the 180 ECTS awarded totally, 15 ECTS are electives; the other 165 ECTS are obtained in mandatory courses. The programme finishes with a bachelor’s thesis of 15 ECTS.

At completion of the 3-year programme a candidate will have achieved a learning outcome defined by the following knowledge, skills and general competence:

Knowledge – the candidate
- has wide knowledge of intelligent systems, applications, architecture, central theories and questions, system development methods and tools, programming IDE
- knows research and development work in the field of intelligent systems
- knows the characteristics and paradigms of intelligent systems
- knows the ecosystem for intelligent systems

Skills – the candidate
- is able to use gained knowledge and relevant results from research and development work on practical and theoretical problems and give reasons for choices taken
- is able to reflect on her/his own work and adjust it under guidance
- masters relevant professional tools (IDE, version control, project management and testing) and techniques
- is able to design and implement solutions for intelligent systems from business concept to solution

General competence – the candidate
- knows relevant ethical issues related to the profession as well as to the academic study
- is able to exchange points of view and experiences with other people with the same background, thus contributing to the development of sound practice
- is acquainted with new thinking and innovation processes

1.3 Central themes

The bachelor’s programme in Intelligent Systems has the following central themes and research basis:

Programming skills are central all through the study, for client-server and for mobile solutions. The study programme aims to develop an understanding of design and implementation of intelligent systems with additional back systems. The study is based on research anchored in intelligent systems and the Internet of Things. The programme cooperates actively with business and industry, especially with the consulting industry. This industry participates in the teaching through guest lectures and workshops.
2 Individual courses offered to exchange students Autumn 2017

2.1 TEK300 Machine Learning

Norwegian name: Maskinlæring
ECTS credits: 7.5
Area of study: Technology/IT
Language of instruction: English
Programme: Mandatory course Bachelor in IT - Intelligent Systems
Required prerequisites: Basic programming knowledge such as PGR100 Programming 1, PGR101 Programming 2, or equivalent previous knowledge
Recommended prerequisites: Basic database knowledge, for instance DB1100 Databases, or equivalent knowledge
Semester: The course is taught in the 5th semester (Autumn)
Course leader: Tomas Sandnes

Course outline
Working with this course will give students knowledge of the mostly used machine learning techniques. Machine learning makes computers able to learn without being explicitly programmed. In the previous decade machine learning has given us more effective web search, practical speech recognition, and self-steering cars. The students gain theoretical knowledge as well as practical skills in machine learning.

Learning outcome
Knowledge: At completion of the course candidates will
• be able to explain central concepts such as machine learning, data mining and statistical pattern recognition
• understand supervised learning, including neural networks
• understand unsupervised learning, including deep learning
• know best practices for machine learning, for instance in bias or variance theory
Skills: At completion of the course candidates will be able to
• apply the most relevant machine learning algorithms
• effectively map data using machine learning
• judge the yield of different machine learning algorithms and compare them
General competence: At completion of the course candidates will
• master machine learning as a tool to identify and make effective use of information
• be able to critically assess existing research in the field of machine learning

Teaching and learning methods
Lectures, exercises and self-study
Recommended workload
Participation in lectures and tutorials – 48 hours
Self-study – 128 hours
Examination and preparing for the examination – 24 hours
**Total recommended workload – 200 hours**

Learning material/Syllabus
Updated information on required reading and other learning material is posted per programme on our electronic learning platform before the semester starts. The information is also available on our website.

In addition to literature and other learning material, scheduled teaching and other scheduled learning activities are part of the syllabus.

Coursework requirements
None

Assessment
Assessment is based on a 3-hour individual written examination. No aids permitted.

Grading scale: A-F with A as the best grade and E as the lowest pass grade. F means failed.

Assessment criteria
See Learning outcome

2.2 INS300 Data Science

**Norwegian name:** Data science

**ECTS credits:** 7.5

**Area of study:** Technology/IT

**Language of instruction:** English

**Programme:** Mandatory course for Bachelor in IT – Intelligent Systems; optional course for Bachelor in IT – E-Business

**Required prerequisites:** None

**Recommended prerequisites:** Databases

**Semester:** The course is taught in the 5th semester (Autumn)

**Course leader:** Wanda Presthus

**Course outline**
Data Science introduces the way different types of data can be compiled, analysed and visualised. This includes statistic knowledge and data mining techniques. Visualisation means to decide how data should best be presented, and an evaluation of how information should be communicated to the various recipients. The students learn theory and gain practical experience with leading tools for analysis and visualisation.
Learning outcome

Knowledge: At completion of the course candidates will
- be able to define data science as a concept
- know various data sources such as database, social media and sensors
- be able to describe basic principles of visualisation
- know principles for presenting information

Skills: At completion of the course candidates will be able to
- master tools for predictive or advanced analysis
- master tools for visual presentation of information and organise it for an end user
- deal with selected techniques in statistics and analysis, such as correlation, regression analysis, clustering, and what-if scenarios

General competence: At completion of the course candidates will be able to
- understand how the entire process from collection of raw data to data visualisation creates value
- critically adapt different presentation formats to message and recipient
- judge ethical issues

Teaching and learning methods

The course is taught over one semester with lectures and exercises.

Recommended workload

Participation in lectures and tutorials – 44 hours
Self-study – 90 hours
Independent exercises and practical work individually or in groups – 63 hours
Examination and preparing for the examination – 3 hours
Total recommended workload – 200 hours

Technology and tools

SAS Visual Analytics and RapidMiner

Learning material/Syllabus

Updated information on required reading and other learning material is posted per programme on our electronic learning platform before the semester starts. The information is also available on our website.

In addition to literature and other learning material, scheduled teaching and other scheduled learning activities are part of the syllabus

Coursework requirements

None
Assessment
Assessment is based on an individual written examination lasting 3 hours. No aids permitted.

Grading scale: A – F with A as the best grade and E as the lowest pass grade. F means failed.

Assessment criteria
See Learning outcome

2.3 TEK200 Machine-to-Machine Communication

Norwegian name: Maskin-til-maskin-kommunikasjon
ECTS credits: 7.5
Area of study: Technology/IT
Language of instruction: English
Programme: Mandatory course for Bachelor in IT – Intelligent Systems
Required prerequisites: Successful completion of TK1100 Digital Technology, or equivalent previous knowledge
Recommended prerequisites: PG5500 Embedded Systems, or equivalent previous knowledge
Semester: The course is taught in the 5th semester (Autumn)
Course leader: Tomas Sandnes

Course outline
The course will give students knowledge of the most essential challenges, solutions and applications for machine-to-machine communication (M2M communication). M2M communication deals with communication processes that do not involve humans, and that usually perform automated tasks. The students will be able to integrate the technology into existing infrastructure. Through the course the students learn to know and use the components of modern M2M solutions.

Learning outcome
Knowledge: At completion of the course candidates will
- be able to define the elements making up an M2M solution
- know and be able to identify the differences between the most important standards (such as REST), protocols (such as MQTT) and algorithms connected with M2M communication
- be able to choose adequate software and components for an M2M system

Skills: At completion of the course candidates will be able to
- set up an M2M system with sensors, actuators and controllers
- solve various challenges in connection with the implementation of M2M communication, both to build new solutions and to integrate the technology into existing systems
- use M2M communication as a tool to improve the efficiency of existing solutions, among other things by identifying bottle necks

General competence: At completion of the course candidates will
- have gained experience with the way modern M2M systems can contribute to innovation
- be able to critically assess existing research related to M2M communication

Teaching and learning methods
Lectures, exercises and self-study

Recommended workload
Participation in lectures and tutorials – 48 hours
Self-study – 104 hours
Examination and preparing for the examination – 48 hours
**Total recommended workload – 200 hours**

Technology and tools
No particular requirements

Learning material/Syllabus
Updated information on required reading and other learning material is posted per programme on our electronic learning platform before the semester starts. The information is also available on our website.

In addition to literature and other learning material, scheduled teaching and other scheduled learning activities are part of the syllabus.

Coursework requirements
The requirement for taking the examination is that the candidate has submitted one mandatory assignment and received approval.

Assessment
Assessment is based on an individual project examination lasting 2 weeks. The assignment consists of building nodes for a sensor network and make the nodes communicate with the surroundings via server services and each other. The assignment requires work with electronics, embedded software, and server services.

Assessment criteria
See Learning outcome

Notes
An ongoing curriculum revision will probably result in adjustments to the curriculum, also concerning the examination. The revised curriculum will be published in the autumn of 2016.
2.4 PRO300 Virtual Reality Project

Norwegian name: Virtual Reality-prosjekt
ECTS credits: 7.5
Area of study: Technology/IT
Language of instruction: English
Programme: Optional course in Bachelor in IT – Programming; Bachelor in IT – Game Programming; Bachelor in IT – Game Design; Bachelor in IT – 3D Graphics; Bachelor in IT – Interactive Design; Bachelor in IT – Intelligent Systems

Required prerequisites: Basic qualifications in programming, interaction, game or concept development, or 3D design

Recommended prerequisites: As above

Semester: The course is taught in the 5th semester (Autumn)

Course leader: Kim Baumann Larsen

Course outline
The Virtual Reality Project is a cross-disciplinary course where students participate, bringing with them their different backgrounds. The purpose of the project is to develop interactive virtual reality applications for games, art or visualisation. The students will make concepts, design and create interactive Virtual Reality solutions, and thereby understand the technological frames in use.

Learning outcome
Knowledge: At completion of the course candidates will
- know the machine and software platforms for Virtual Reality
- know the platform requirements for different Virtual Reality solutions
- know the possibilities and limitations with Virtual Reality
- know the demands for achieving the largest possible presence in a Virtual Reality solution
- know basic principles for storytelling for interactive Virtual Reality

Skills: At completion of the course candidates will be able to
- draft and plan one interactive Virtual Reality concept for games, art or visualisation
- make prototypes for different platforms
- design and implement one interactive Virtual Reality experience for a chosen platform
- integrate 3D models, sound elements and interaction points into a holistic Virtual Reality experience for a chosen platform

General competence: At completion of the course candidates will be able to
- assess and create an optimal Virtual Reality solution for a chosen platform
- assess different Virtual Reality platforms and the possibility and limitations of solutions
Teaching and learning methods
Lectures and exercises, and one cross-disciplinary project in which students participate with their different study programme backgrounds either in programming, interaction, games/concept development, or 3D design.

Recommended workload
Participation in lectures and tutorials – 25 hours
Self-study – 80 hours
Independent preparation for presentations or discussions in class – 5 hours
Student work with projects, productions, assignments etc. – 20 hours
Independent exercises, lab work, practical work individually or in groups – 70 hours
Total recommended workload – 200 hours

Technology and tools
HTML editor

Learning material/Syllabus
Updated information on required reading and other learning material is posted per programme on our electronic learning platform before the semester starts. The information is also available on our website.

In addition to literature and other learning material, scheduled teaching and other scheduled learning activities are part of the syllabus.

Coursework requirements
None

Assessment
Assessment is based on a combined examination consisting of the following elements:

- A group project examination (60%) where the groups submit the results of the project they have been working with in the course
- A group written examination (20%) consisting of a group report describing how the project was carried out, as well as reflections in connection with the work (a process document). The report should have a length of 3000-5000 words. The specifications of requirements for the report are handed out three weeks before submission.
- A group oral examination (20%) in which the groups present the project result and the process. The oral examination lasts approx. 25 minutes.

Grading scale: A-F with A as the best grade and E as the lowest pass grade. F means failed.
2.5 PG3300 Software Design

Norwegian name: Software Design
ECTS credits: 75
Area of study: Technology/IT
Language of instruction: English
Programme: Mandatory for Bachelor in IT – Programming and Bachelor in IT – Game Programming; optional course in Bachelor in IT – Intelligent Systems

Required prerequisites: Experience with basic object oriented programming, such as PGR100 Programming 1 and PGR101 Programming 2, or equivalent knowledge
Recommended prerequisites: None
Semester: The course is taught in the 3rd or 5th semester (Autumn)
Course leader: Tomas Sandnes

Course outline
The course will make students able to design and further develop extensive software systems using well-known techniques for modelling, testing and implementation.

Learning outcome
Knowledge: At completion of the course candidates will
- know the background and content for the UML standard
- know what Unit Testing is
- be able to explain the principles of test-driven development (TDD)
- know par programming and how the use of it influences software projects
- know what design patterns are
- know and be able to identify some important design patterns when reading them in code
- know what refactoring is
- know what multi-threading is
- understand how locking can be used to code thread securely
- be able to explain the principles of event handling

Skills: At completion of the course candidates will be able to
- know and use the basic syntax in the programming language C#, and know in what ways it is different from Java
- handle the UML diagrams: use case, class diagram, and sequence diagram
- use UML to design program architecture
- participate productively in par programming
- implement the following patterns: MVC & MVP, singleton, factory, builder, flyweight, composition, decorator
- use design guidelines such as layers and the GRASP principles emphasising the following for the latter: controller, information, expert, low coupling, high cohesion
- implement applications that apply several threads
- master an integrated development tool (IDE)
- write and edit source code with the mentioned tool

General competence: At completion of the course candidates will be able to
• reflect on multi threading and its usage
• cooperate with other programmers to develop good programs and to further develop their competence
• assess the quality of existing programs and possible structural improvements

Teaching and learning methods
Lectures, exercises, self-study

Recommended workload
Participation in lectures and tutorials – 48 hours
Self-study – 104 hours
Examination and preparing for the examination – 48 hours
Total recommended workload – 200 hours

Technology and tools
IDE: Visual Studio

Learning method/Syllabus
Updated information on required reading and other learning material is posted per programme on our electronic learning platform before the semester starts. The information is also available on our website.

In addition to literature and other learning material, scheduled teaching and other scheduled learning activities are part of the syllabus.

Coursework requirements
None

Assessment
Assessment is based on a two-part examination consisting of the following elements:

• A written examination in groups (40%) for which the candidates submit self-developed program code and an accompanying document. The assignment has a time span of 3 weeks.
• An individual oral examination (60%) of 20 minutes. The oral examination takes its point of departure in the above-mentioned work (the written examination), and is used to decide an individual final grade adjusted according to the individual candidate’s performance as shown in the oral examination situation.

Assessment criteria
See Learning outcome
2.6 PG5600 iOS Programming

**Norwegian name:** iOS-programmering

**ECTS credits:** 7.5

**Area of study:** Technology/IT

**Language of instruction:** English

**Programme:** Mandatory course for Bachelor in IT – Mobile Programming, optional for Bachelor in IT – Intelligent Systems

**Required prerequisites:** Advanced knowledge of programming (from 2nd year Bachelor in IT) or equivalent previous knowledge in object oriented programming

**Recommended prerequisites:** None

**Semester:** The course is taught in the 5th semester (Autumn)

**Course leader:** Tor-Morten Grønli

**Course outline**

The course gives an introduction to programming in Swift and the iOS platform. At completion of the course the students will be able to make applications that communicate over networks, store data locally, and use interface elements and patterns that go with the platform.

**Learning outcome**

**Knowledge:** At completion of the course candidates will

- have gained knowledge of the architecture of the iOS platform
- be able to describe the life cycle of an iOS application using text and drawing
- know the process of distributing applications
- know how applications are compiled on iOS
- be able to describe MVC, the observable and delegate pattern in context with iOS, using text and drawing

**Skills:** At completion of the course candidates will be able to

- program basic Swift, including
  - data types
  - use of foundation classes
  - control structures
  - use of object orientation
  - use of protocols
  - use of extensions
  - use of closures
  - error handling
- use iOS APIet actively when programming applications
- apply the usual GUI components to produce user interfaces
- use storyboards
- write automated texts
- make use of basic animations
- serialise, de-serialise and persist data
• programme to internet-based services
• account for and implement asynchronous architecture
• debug and deploy an application to simulator and unit

General competence: At completion of the course candidates will be able to
• propose and give reasons for the choice of architecture in an iOS application
• know when asynchronism is relevant
• consume a REST API

Teaching and learning methods
The course is taught with 12 lectures, each of about 2 hours, and 12 teacher-led exercises of about 2 hours each. The teacher-led exercises are not mandatory, but to obtain the competence goals the students are expected to complete exercises and additionally put in some extra effort if the teacher-led exercises are not sufficient.

Recommended workload
Participation in lectures and tutorials – 48 hours
Self-study – 110 hours
Examination and preparing for the examination – 42 hours
Total recommended workload – 200 hours

Technology and tools
iOS SDK (inkl Xcode)
iOS

Learning material/Syllabus
Updated information on required reading and other learning material is posted per programme on our electronic learning platform before the semester starts. The information is also available on our website.

In addition to literature and other learning material, scheduled teaching and other scheduled learning activities are part of the syllabus.

Coursework requirements
None

Assessment
Assessment is based on an individual written examination (100%) in which candidates submit an extensive examination paper. The paper tests knowledge as well as skills. The assessment criteria for the examination paper are handed out together with the question paper 2-3 weeks before submission.

Grading scale: A-F with A as the best grade and E as the lowest pass grade. F means failed.

Assessment criteria
See Learning outcome