

<b>DEGREE PROFILE OF</b> <b>Bachelor í KT-verkfrøði</b> <i>Bachelor of Science in Software Engineering</i>
<i>2. Edition – 27. February 2017 / Hannes Gislason</i>

TYPE OF DEGREE & LENGTH	<b>Single degree</b> (180 ECTS/3 years). About 70% of the education is provided locally by the University of the Faroe Islands and about 30% of the education is provided as distance education in collaboration with the Mid Sweden University, Sweden.
INSTITUTION(S)	Fróðskaparsetur Føroya, Megindeildin fyri náttúruvísind og heilsuvísind, Náttúruvísindadeildin. <i>University of the Faroe Islands, Faculty of Natural and Health sciences, Department of Science and Technology.</i>
ACCREDITATION ORGANISATION(S)	The Ministry of Education, Research and Culture (MMR), The Faroe Islands. Bi-annual student intake approved since 2010. The next student intake is planned for 2018.
PERIOD OF REFERENCE	Degree programme starting 2016; curriculum 2016-2019.
CYCLE /LEVEL	National level: Bachelor; QF for EHEA: 1st cycle; EQF level 6.

<b>A</b>	<b>PURPOSE</b>
	The purpose of the programme is to educate software engineers who can apply scientific principles – mainly from computer science and mathematics - in order to provide and maintain effective, fast and reliable software systems to organizations, businesses and the public administration. The bachelor programme is also intended to provide a solid basis for further studies and scientific research within computer science, software engineering or a related discipline in the rapidly evolving field of Information and Communication Technologies (ICT).

<b>B</b>	<b>CHARACTERISTICS</b>	
1	DISCIPLINE(S) / SUBJECT AREA(S)	The main disciplines are: mathematics and statistics, computer science and programming; general engineering courses and projects (13:62:25): <ul style="list-style-type: none"> <li>- Mathematics and statistics (22.5 ECTS, 12.5%)</li> <li>- Computer Science and programming (112.5 ECTS, 62.5%)</li> <li>- General engineering courses and projects (45 ECTS, 25%)</li> </ul>
2	GENERAL / SPECIALIST FOCUS	The <b>general focus</b> is on a broad introduction to computer science, discrete mathematics, data science and statistics, data communications and networks, algorithms and data structures, operating systems, databases, web and mobile applications, software security, engineering economics and project management. The <b>specialist focus</b> is on software development, which is mainly based on a progression of courses in the C++ language. The students are progressively specialized from learning basic procedural programming to learn object oriented programming, and finally to master design patterns and larger software engineering projects.
3	ORIENTATION	The orientation of the bachelor degree programme in software engineering is both scientific - with basic computer science and mathematics courses - and applied with the programming courses. The specialist focus on software development is mainly

		applied. The choice of the bachelor project is based on the individual student interests and on the availability of potential supervisors from the IT-industry, from the University of the Faroe Islands or from other universities. Depending on interests and possibilities, the student may either choose an applied project or a basic scientific research project.
4	DISTINCTIVE FEATURES	<ul style="list-style-type: none"> <li>- Close collaboration with professionals from the local Faroese IT-industry who teach some courses and supervise the students during their bachelor projects.</li> <li>- About 30% of the education is provided as distance education in collaboration with the Mid Sweden University, Sweden.</li> <li>- The local teaching language is mainly Faroese and English. Textbooks are mostly English and occasionally Danish. Swedish language may occasionally be used in exercises, notes and correspondence from Mid Sweden University during the distance courses, but the textbooks are English.</li> </ul>

<b>C EMPLOYABILITY &amp; FURTHER EDUCATION</b>		
1	EMPLOYABILITY	<ul style="list-style-type: none"> <li>- IT System designer/developer.</li> <li>- Web- and mobile applications designer/developer</li> <li>- IT-manager, operations- or project manager, Data analyst.</li> <li>- Research and/or teaching.</li> </ul>
2	FURTHER STUDIES	<p>Software Engineering, Computer Science and related fields. Access to DTU-MSc: Computer Science and Engineering, Digital Media Engineering, Telecommunication; DIKU-MSc Computer Science; MIUN-Intern. MSc. in Computer Engineering. Previous students have also been accepted for further software- and computer science studies at ITU.DK and KU.dk.</p> <p>Further studies at other universities – possibly with some conditions/restrictions - may also be possible.</p>

<b>D EDUCATION STYLE</b>		
1	LEARNING & TEACHING APPROACHES	Teacher centred class room lectures; student centred exercises to be solved in class and/or handed in, group- and individual work with computer exercises and projects. The distance education is teacher guided with remote instructors, but it may be supported by local instructors.
2	ASSESSMENT METHODS	Examples of the most used examination forms are: written examinations (that count 100%); written examinations (70%) and hand-ins (30%); oral examinations (100%). Distance education: a) hand-ins 3 ECTS, written examination 1.5 ECTS, project 3 ECTS; b) hand-ins 4.5 ECTS and written examination 3 ECTS. In some cases, other examinations-forms may be used.

<b>E</b>	<b>PROGRAMME COMPETENCES</b>
1	GENERIC
	<p><b>The profession:</b> understand the software engineering knowledge area and the software engineering profession.</p> <p><b>Analytical:</b> ability to apply abstract and analytical thinking in synthesis of new ideas.</p> <p><b>Management:</b> ability to design and plan, manage time and resources in engineering projects.</p> <p><b>Team-work:</b> ability to work in a team of designers/developers in engineering projects.</p> <p><b>Research:</b> ability to undertake small research projects at the undergraduate level.</p> <p><b>International:</b> ability to work across national borders and to communicate in the English language.</p>
2	SUBJECT SPECIFIC
	<p><b>Mathematical Ability:</b> ability to apply discrete mathematics and lin. algebra in the design and use of algorithms and data structures. <b>Data Analysis Ability:</b> ability to understand and apply data science concepts, statistical concepts and statistical software to data analysis. <b>Technological Ability:</b> strong capacity to understand and learn, stay-up-to-date with, and use information and communication technologies. <b>Computing Ability:</b> strong capacity to analyse, model, design, program and implement software systems. <b>Problem-solving Ability:</b> strong ability to identify and resolve problems in software or hardware systems.</p>

F	COMPLETE LIST OF PROGRAMME LEARNING OUTCOMES
	<ol style="list-style-type: none"> <li>1. Profession: master software engineering skills and knowledge, necessary to begin a career in the IT-industry or as the basis for further studies in software engineering, computer science, data science or a related field of study within information and communication technologies.</li> <li>2. Mathematical knowledge: ability to understand both basic and moderately advanced concepts from discrete mathematics and linear algebra, and ability to apply these concepts to computer science.</li> <li>3. Computer Science knowledge: obtain a broad and deep understanding of computer science, data communications, wired and wireless networks, the internet, web- and mobile applications, databases, security, computer architecture and operating systems, algorithms and data structures, corporate IT-systems.</li> <li>4. Basic programming skills: obtain good ability to design and develop simple procedural programmes based on the C++ language. Understand the differences between C and the C++ language.</li> <li>5. Object Oriented Programming: good ability and routine to design and develop moderately advanced programmes based on object oriented design and the C++ language. Also know the differences between the C++ language, Java and C#.</li> <li>6. Design Patterns: demonstrate ability to design and develop advanced programmes based on object oriented design, design patterns, the UML-modelling language and the re-usability of programming code.</li> <li>7. Web and mobile programming: demonstrate both broad and deep knowledge of web- and mobile applications development, web design, web servers and clients, web-stores, apps and E-business. Ability to design and implement web- and mobile applications based on both open source script languages and corporate object based frameworks like ASP.NET with MVC/C#. Further, demonstrate the ability to develop simple Apps, e.g. for the Android-platform.</li> </ol> <ol style="list-style-type: none"> <li>1. Data analysis skills: obtain basic knowledge of data science, statistical theory and probability with applications to data analysis, research and computer science, e.g. analysis of network traffic, web logs, user surveys etc. Obtain basic knowledge and mastery of the statistical software package R/RStudio. Be introduced to modern data science, statistics and the concept of reproducible research by using advanced - but easy to use - facilities in RStudio. Be introduced to data science and statistics within computer science and other fields, e.g. human health, biology, and sociology.</li> <li>2. Software Security: ability to understand both basic and advanced concepts in software security and to perform risk analysis of software systems. Ability to design, implement and maintain secure software systems.</li> <li>3. Engineering Economics: ability to understand engineering economics, to make budgets and to perform financial control.</li> <li>4. Project management: ability to plan and lead software engineering projects, and to manage resources in order to develop and deliver software projects within expected time limits and costs estimates.</li> <li>5. Research skills: understand how research is performed, presented and published. Ability to search and critically read textbooks and scientific literature. Understand different literature sources, both primary and secondary sources, how to cite them properly and how to avoid plagiarism. Understand,</li> </ol>

	<p>construct and explore basic and applied research questions in order to produce new knowledge and/or new applications.</p> <p>13. Communication skills: ability to document programming code, ability to write project proposals, ability to document and share progress during project work, ability to write project reports and to make oral presentations of the projects results.</p> <p>14. Project work: demonstrate and combine computer science knowledge, software engineering skills, and research- and communication skills in project work.</p>