



HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

**Learning Outcomes in The School of
Computer Science at
Reykjavik University**

**Computer Science
Software Engineering
Mathematics**

1st of March 2007

Efnisyfirlit

COMPUTER SCIENCE	3
Diploma in Applied Computing (Cycle 1, level 1)	3
BSc in Computer Science (Cycle 1, level 2)	5
MSc in Computer Science (Cycle 2, level 4)	8
SOFTWARE ENGINEERING	13
BSc Software Engineering (Cycle 1, level 2)	13
MATHEMATICS	17
BSc in Mathematics (Cycle 1, level 2)	17

Computer Science

Diploma in Applied Computing (Cycle 1, level 1)

Diploma in Applied Computing in the School of Computer Science is a two years (four semesters) programme. The total credits are 120 ECTS. Emphasis is on Application Software, design and programming. Each semester students take five courses (6 ECTS).

On the completion of graduate study at cycle 1, level 1, the following criteria shall be fulfilled:

Knowledge and understanding

At the end of the degree course, a student will possess a general understanding of the following topics.

- The mathematical foundations underpinning computer science. This includes knowledge of topics from discrete mathematics, basic logic, basic probability theory and statistics.
- Algorithms and data structures
- Object-oriented programming in C++ and Java.
- The fundamental principles of von Neumann computer architectures, operating systems and distributed systems.
- The fundamental principles of data modelling and database design, with emphasis on relational database concepts.
- The principles of software engineering with emphasis on object-oriented system analysis and design, and UML.
- The fundamentals and basic theory of human computer interaction and usability testing.
- The fundamental principles of internet technology and Windows programming.
- The principles and techniques from the elective areas in which the student decided to develop special expertise. Such expertise is developed by following elective courses, by means of independent studies or during the end of term project work. (See the course catalogue for a detailed description of the available options, together with their expected learning outcomes.)

Type of knowledge

Students have adopted extensive knowledge and understanding in the following specialized areas.

- Object-oriented programming, with focus on C++ and Java.
- Databases, with emphasis on query languages like SQL, object oriented databases, query processing and optimization.
- Programming with the .NET environment.
- User interface design and evaluation, with focus on Windows systems and web-based systems.
- Operating systems, with emphasis on Unix/Linux and Windows XP operating systems.

Practical Skills

- Analyze the requirements, design, implement, test, document, and maintain a computer-based system.
- Work as a member of a team, contributing to the management, planning and implementation of a computer system.
- Make use of a range of theories, techniques and tools from programming languages, programming environments, and system analysis and design.
- Independently propose a system development project, plan its execution, undertake its development, evaluate its outcome, and report on its results in a professional manner.

Theoretical skills

- Analyze a problem, consider a range of possible approaches to its solution, determine the most promising approaches, and select one for its computer-based implementation.
- Evaluate the results of a computer-based solution to a problem, and propose possible improvements.
- Develop consistent, complete and short descriptions of the requirements for computer-based systems, of their designs, and of the choices made in their actual implementation.

Communication skills and information literacy

- Propose, plan and manage team projects using support tools, e.g., for configuration management.
- Work effectively as a team, and be able to manage conflicts.
- Communicate effectively and professionally both in writing and by means of presentations using appropriate technical language.
- Find information that is relevant to the student's task using search engines, on line libraries and repositories.
- Use facilities for the professional production of presentations and written documents. These include, e.g., typesetting systems, editors, spell checkers, etc.
- Be able to manage a team of individuals working on a development project.

Learning Skills

- Solve problems using the acquired skills or knowledge.
- Use known information, methods, concepts and theories in new situations.
- Master the material learned during the degree course.
- Interpret facts by comparing them and contrasting them with one another.
- Infer possible causes from the available data.
- Use known facts to create new ones,.
- Make choices based on reasoned arguments, and evaluate the outcomes of those choices by comparing them with alternative solutions.
- They have developed personal autonomy and independence necessary to undertake further study at higher levels of education, f.ex. at the BSc level.

BSc in Computer Science (Cycle 1, level 2)

Traditional study in the School of Computer Science is a three year (six semesters) BS programme. The total credits are 180 ECTS.

On the completion of graduate study at cycle 1, level 2, the following criteria shall be fulfilled, in addition to the criteria fulfilled at former level(s):

Knowledge and understanding
<p>At the end of the degree course, a student will possess a general understanding of the following topics.</p> <ul style="list-style-type: none">• The mathematical foundations underpinning computer science. This includes knowledge of topics from calculus, discrete mathematics, basic logic, basic probability theory and statistics.• Algorithmics, with special emphasis on the analysis and design of efficient algorithms and data structures for solving real-world problems.• Theory and practice of programming languages, with special emphasis on the principles of object-oriented programming languages.• The fundamental principles of von Neumann computer architectures, operating systems and distributed systems.• The fundamental principles of data modelling and database design, with emphasis on relational database concepts.• The principles of software engineering with emphasis on object-oriented system analysis and design, and UML.• Project management and quality control in software development, legal aspects of contracts and methodologies for configuration management.• The fundamentals and basic theory of human computer interaction and usability testing.• The fundamental principles of internet technology and Windows programming.• The principles and techniques from the elective areas in which the student decided to develop special expertise. Such expertise is developed by following elective courses, by means of independent studies or during the end of term project work. (See the course catalogue for a detailed description of the available options, together with their expected learning outcomes.)

Type of knowledge

Students have adopted extensive knowledge and understanding in the following specialized areas.

- Object-oriented programming, with focus on C++ and Java.
- Databases, with emphasis on query languages like SQL, object oriented databases, query processing and optimization.
- Programming with the .NET environment.
- User interface design and evaluation, with focus on Windows systems and web-based systems.
- Operating systems, with emphasis on Unix/Linux and Windows XP operating systems.
- Advanced topics from the elective areas in which the student decided to develop special expertise. Such expertise is developed by following elective courses, by means of independent studies or during the end of term project work. (See the course catalogue for a detailed description of the available options, together with their expected learning outcomes.) Sample such topics are game design and programming, internet protocols, methods and tools for the formal modelling and verification of computing systems, search methods in artificial intelligence, and virtual environments.

Practical Skills

- Analyze the requirements, design, implement, test, document, and maintain a computer-based system.
- Work as a member of a team, contributing to the management, planning and implementation of a computer system.
- Make use of a range of theories, techniques and tools from programming languages, programming environments, and system analysis and design.
- Be able to search for relevant literature related to an area of study, and to evaluate its contents and relative importance.
- Independently propose a system development project, plan its execution, undertake its development, evaluate its outcome, and report on its results in a professional manner.

Theoretical skills

- Analyze a problem, consider a range of possible approaches to its solution, determine the most promising approaches, and select one for its computer-based implementation.
- Evaluate the results of a computer-based solution to a problem, and propose possible improvements.
- Develop consistent, complete and short descriptions of the requirements for computer-based systems, of their designs, and of the choices made in their actual implementation.
- Abstract the essential core of a problem, and realize the commonalities between seemingly different tasks to be accomplished.
- Reason at a level of abstraction that is appropriate while pursuing the solution to a technical problem.

Communication skills and information literacy

- Propose, plan and manage team projects using support tools, e.g., for configuration management.
- Work effectively as a team, and be able to manage conflicts.
- Communicate effectively and professionally both in writing and by means of presentations using appropriate technical language.
- Find information that is relevant to the student's task using search engines, on line libraries and repositories.
- Use facilities for the professional production of presentations and written documents. These include, e.g., typesetting systems, editors, spell checkers, etc.
- Analyze complex real-world problems and devise efficient and well-documented computer-based solutions for those.
- Use mathematical models and their associated analysis techniques in the design and evaluation of solutions for problems.
- Be able to manage a team of individual s working on a development project.

Learning Skills

- Solve problems using the acquired skills or knowledge.
- Use known information, methods, concepts and theories in new situations.
- Generalize from a collection of specific instances.
- Draw conclusions and predict possible outcomes.
- Discover patterns in the available information.
- Master the material learned during the degree course.
- Interpret facts by comparing them and contrasting them with one another.
- Infer possible causes from the available data.
- Use known facts to create new ones,.
- Make choices based on reasoned arguments, and evaluate the outcomes of those choices by comparing them with alternative solutions.
- They have developed personal autonomy and independence necessary to undertake further study at higher levels of education, f.ex. at the MSc level.

MSc in Computer Science (Cycle 2, level 4)

The M.Sc. program is a two year (120 ECTS) graduate program in Computer Science. Students may either choose the research-based track, where 60 ECTS are devoted to courses and 60 ECTS to an individual research project, or the course-based track, where 90 ECTS are devoted to courses and 30 ECTS to an MSc project, which may be a group project. The study program is closely tied to the research carried out at the Department of Computer Science, through research-based courses and advanced research projects. The goal of the program is to prepare students for leading careers in industry and/or further academic study.

On the completion of graduate study at master level in addition to first cycle studies, the following criteria shall be fulfilled, in addition to the criteria fulfilled at former level(s):

Knowledge and understanding

At the end of the degree course, a student will possess a solid understanding and knowledge of the following topics.

- Basic theoretical principles in Computer Science. This includes knowledge of
 - various types of finite automata,
 - the formal definitions of programming languages and their connection with automata,
 - Turing machines and computability theory, and
 - algorithmic complexity classes.
- Research methodology, including basic history of science, the fundamentals of scientific writing, how to give a scientific talk, how to evaluate a scientific paper, and research ethics.
- Statistical principles, and software tools embodying those.
- Advanced principles and techniques from the elective areas in which the student decided to develop special expertise. Such expertise is developed by following elective courses in the research areas of the members of staff, by means of advanced independent studies, and mainly during the MSc thesis work. Areas of specialization include artificial intelligence (e.g., agent technology, computer games, robotics and virtual environments), concurrency theory (with emphasis on modelling and verification of reactive systems, process algebra, and structural operational semantics), databases (with focus on efficient indexing of multimedia databases), and language technology (e.g., tagging of Icelandic and software support for the analysis of Icelandic text).
- Established and potential applications of techniques developed within the chosen area of specialization.

Type of knowledge

- Solid knowledge and understanding of the theory and practical skills introduced during the courses.
- State-of-the-art knowledge in the research area within which the students have carried out research leading to their MSc theses.
- Thinking like a researcher in computer science at the very early stage of her/his own career.

Practical Skills

- Analyze the requirements, design, implement, test, document, and maintain a computer-based system.
- Work as a member of a team, contributing to the management, planning and implementation of a computer system.
- Make use of a range of theories, techniques and tools from programming languages, programming environments, and system analysis and design.
- Be able to search for relevant literature related to an area of study, and to evaluate its contents and relative importance.
- Independently propose a small scale research project, plan its execution, undertake its development, evaluate its outcome, and report on its results in a professional manner.
- Be able to place the results of their work in the context of the research literature on the subject on which the students choose to specialize.

Theoretical skills

- Analyze a well defined, research level problem, consider a range of possible approaches to its solution, determine the most promising approaches, and select one for its (computer-based) implementation.
- Understand the specific characteristics of scientific research, its progress, milestones and ethical values that make it the source of reliable knowledge.
- Evaluate a solution to a well defined, research problem, and propose possible improvements.
- Develop consistent, complete and concise descriptions of the requirements for a satisfactory solution to a well defined, research level problem in computer science.
- Abstract the essential core of a research problem, and realize the commonalities between seemingly different tasks to be accomplished, and techniques and results from different research areas in computer science.
- Reason at a level of abstraction that is appropriate while pursuing the solution to a research problem.
- Be familiar with research methods, techniques, and problem solving approaches from the field of research in which they are specializing.
- Be able to inferring new knowledge from the collected data.

Communication skills and information literacy

- Propose, plan and manage well defined research projects involving a team of individuals.
- Work effectively as a member of a team, and be able to manage conflicts.
- Communicate effectively and professionally both in writing and by means of presentations using appropriate technical language.
- Find information that is relevant to the student's task using search engines, on line libraries and repositories.
- Use facilities for the professional production of presentations and written documents.
- Analyze complex real-world problems and devise efficient and well-documented computer-based solutions for those.
- Use mathematical models and their associated analysis techniques in the design and evaluation of solutions for problems.
- Be able to make an informed use of software tools embodying the most recent theoretical advances in their area of specialization.
- Be able to manage a team of individuals working on a small scale research project.
- Be able to analyze statistical data, and communicate their findings using, e.g., charts and statistical software.
- Be able to report on their work, and that of others, both to a specialist and a general audience.
- Be able to discuss ethical issues in research work with their peers in an informed and reasoned fashion.

Learning Skills

- Solve non-trivial problems independently using the acquired skills or knowledge.
- Master the material learned during the degree course.
- Possess in-depth knowledge of their main area of specialization.
- Be able to ask new questions based on the available information and knowledge.
- Make creative use of known information, methods, concepts and theories in new situations.
- Generalize from a collection of specific instances.
- Identify the key components in the solution to a well defined research level problem.
- Draw conclusions and predict possible outcomes.
- Discover patterns in the available information.
- Interpret facts by comparing them and contrasting them with one another.
- Verify the value of the available evidence.
- Recognize and evaluate subjective opinions.
- Infer possible causes from the available data.
- Use known facts to create new ones.
- Make choices based on reasoned arguments, and evaluate the outcomes of those choices by comparing them with alternative solutions.
- Be able to discuss the results of research work carried out by

themselves or others with their peers.

- They have developed personal autonomy and independence necessary to undertake further study at higher levels of education, f.ex. at the doctorate level.

Software Engineering

BSc Software Engineering (Cycle 1, level 2)

The BSc in Software Engineering is a three year (six semesters) BSc programme. The total credits are 180 ECTS. Each semester students take five courses (6 ECTS). Software systems are among the largest and the most complicated systems ever built. This holds true, in particular, because such systems are increasingly expected to exhibit high reliability. The aim of the Software Engineering degree program is to educate professionals who will apply sound engineering methods in designing and building software-based systems. Most graduates from the program will become software developers, consultants or project leaders in companies developing and selling software-based computing systems. The course offers strong foundations in computer science, mathematics, and basic subjects of engineering. It places great emphasis on formal methods in software system development, as well as on project management and quality control.

Knowledge and understanding

At the end of the degree course, a student will possess a general understanding of the following topics.

- The mathematical foundations underpinning software engineering. This includes knowledge of topics from calculus, discrete mathematics, basic logic, basic probability theory and statistics.
- The theory of finite automata, computability and complexity theory.
- Algorithmics, with special emphasis on the analysis and design of efficient algorithms and data structures for solving real-world problems.
- Theory and practice of programming languages, with special emphasis on the principles of object-oriented programming languages.
- Principles of compiler design.
- The fundamental principles of von Neumann computer architectures, operating systems and distributed systems.
- The fundamental principles of data modelling and database design, with emphasis on relational database concepts.
- The principles of software engineering with emphasis on object-oriented system analysis and design, and modelling languages.
- Project management and quality control in software development, legal aspects of contracts and methodologies for configuration management.
- The fundamental principles of internet technology and GUI programming.
- Basic physics for engineering.
- New technology and its applications.
- Entrepreneurship.
- The principles and techniques from the elective areas in which the student decided to develop special expertise. Such expertise is developed by following elective and/or end of term intensive courses, by means of independent studies or during the end of term project work. (See the course catalogue for a detailed description of the available options, together with their expected learning outcomes.)

Type of knowledge

Students have adopted extensive knowledge and understanding in the following specialized areas.

- Modern object-oriented programming.
- Databases, with emphasis on query languages like SQL, object oriented databases, query processing and optimization.
- Internet programming.
- User interface design and evaluation, with focus on GUI systems and web-based systems.
- Modern operating systems.
- Software engineering and project management.
- Advanced topics from the elective areas in which the student decided to develop special expertise. Such expertise is developed by following elective and/or end of term intensive courses, by means of independent studies or during the end of term project work. (See the course catalogue for a detailed description of the available options, together with their expected learning outcomes.) Sample such topics are game design and programming, internet protocols, methods and tools for the formal modelling and verification of computing systems, search methods in artificial intelligence, and virtual environments.

Practical Skills

- Analyze requirements, design, implement, test, document, and maintain a computer-based system.
- Work as a member of a team, contributing to the management, planning and implementation of a computer system.
- Make use of a range of theories, techniques and tools from programming languages, programming environments, and system analysis and design.
- Be able to search for relevant literature related to an area of study, and to evaluate its contents and relative importance.
- Independently propose a system development project, plan its execution, undertake its development, evaluate its outcome, and report on its results in a professional manner.

Theoretical skills

- Analyze a problem, consider a range of possible approaches to its solution, determine the most promising approaches, and select one for its computer-based implementation.
- Evaluate the results of a computer-based solution to a problem, and propose possible improvements.
- Develop consistent, complete and short descriptions of the requirements for computer-based systems, of their designs, and of the choices made in their actual implementation.
- Abstract the essential core of a problem, and realize the commonalities between seemingly different tasks to be accomplished.
- Reason at a level of abstraction that is appropriate while pursuing the solution to a technical problem.

Communication skills and information literacy

- Propose, plan and manage team projects using support tools, e.g., for configuration management.
- Work effectively as a team, and be able to manage conflicts.
- Communicate effectively and professionally both in writing and by means of presentations using appropriate technical language.
- Find information that is relevant to the student's task using search engines, on line libraries and repositories.
- Use facilities for the professional production of presentations and written documents.
- Analyze complex real-world problems and devise efficient and well-documented computer-based solutions for those.
- Use mathematical models and their associated analysis techniques in the design and evaluation of solutions for problems.
- Be able to manage a team of individual s working on a development project.

Learning Skills

- Solve problems using the acquired skills or knowledge.
- Use known information, methods, concepts and theories in new situations.
- Generalize from a collection of specific instances.
- Draw conclusions and predict possible outcomes.
- Discover patterns in the available information.
- Master the material learned during the degree course.
- Interpret facts by comparing them and contrasting them with one another.
- Infer possible causes from the available data.
- Use known facts to create new ones,.
- Make choices based on reasoned arguments, and evaluate the outcomes of those choices by comparing them with alternative solutions.
- Have the necessary autonomy and independande to be able to continue studies at advanced level i.e. at masters level.

Mathematics

BSc in Mathematics (Cycle 1, level 2)

The BSc degree program at Reykjavik University is an undergraduate study program consisting of 180 ECTS credits, which takes the students 3 years (i.e. six semesters) to complete at a normal pace. The students are required to take at least 120 ECTS credits within mathematics, of which 60 ECTS are obligatory courses. The students can take up to 60 ECTS credits in courses at other departments and institutes of Reykjavik University.

The teaching methods in most of the mathematics courses consist of a mixture of lectures and problem solving sessions, where the students work alone or in small groups, under the guidance of the teacher. The evaluation is based on individual projects/problem sets, which the students hand in throughout the semester, and oral exams, in which their understanding of the solutions they handed in is tested, and they are asked general questions about the main concepts of the given course.

Knowledge and Understanding
<p>At the end of the study program, the student will possess a general understanding of the following topics:</p> <ul style="list-style-type: none">• Sets, functions, relations and induction.• The foundations of abstract algebra, including the basic properties of groups, rings, fields and modules.• Basic properties of the natural numbers, e.g. divisibility, prime numbers, congruence groups, and how these topics are connected to the theory of encryption.• Linear maps, matrices, systems of linear equations and their solutions, determinants, bases of vector spaces, eigenvalues and eigenvectors and the geometric interpretations of these concepts.• The foundations of discrete mathematics, including combinatorics and graph theory.• Limits, derivatives and integrals, along with their definitions and applications in one and several variables.• The foundations of probability theory and statistics, probabilities, probability spaces, events, distributions and the application of these concepts.• Numerical and symbolic methods in mathematical computation.• The main concepts, results and methods from other chosen fields of mathematics or other subjects.
Type of Knowledge
<p>At the end of the BSc program the student should possess a basic knowledge of all the fields of modern mathematics, that is:</p> <ul style="list-style-type: none">• Calculus in One and Several Variables• Linear and Abstract Algebra• Probability Theory and Statistics• Discrete Mathematics <p>Additionally the students will have deepened their knowledge in one or more of the above subfields and familiarized themselves with at least one of the following fields of abstract mathematics:</p> <ul style="list-style-type: none">• Topology• Number Theory• Graph Theory• Combinatorics• Optimization• Measure and Integration Theory• Complex Analysis

Practical Skills

At the end of the degree course the students should be able to

- Write clear and well-organized mathematical texts, which can be used by others
- Write and use computer programs in order to test hypotheses and models in mathematics and other fields
- Analyze and process statistical data and be able to formulate an objective opinion about the work of others in this area
- Apply mathematics to practical problems
- Put forward a hypothesis using mathematical language, design a plan for testing the hypothesis and carry it out
- Work independently on more substantial problems that are presented to them
- Take up a professional career in fields that require knowledge of science and logical reasoning

Theoretical Skills

At the end of the degree course the students should be able to

- Read a well-organized mathematical text
- Evaluate the quality of mathematical reasoning of other people
- Apply the results of other scientists in their own work

Communication skills and information literacy

At the end of the degree course the students should be able to

- Work in groups in order to solve mathematical problems
- Communicate their thoughts on mathematics to others and take part in a two-way discussion
- Work independently and understand a clear mathematical text without assistance
- Recognize the main thoughts and concepts behind mathematical definitions, theorems and proofs and be able to explain these observations to others
- Give a lecture, where they present their results and conclusions
- Search for information and data and assess its credibility
- Write mathematical articles using mathematical typesetting software
- Contribute to a democratic debate as experts in fields where knowledge of mathematics, science or statistics is required

Learning skills

At the end of the degree course the students should be able to

- Take on a further, post-graduate research-based study program in mathematics
They have developed personal autonomy and independence necessary to undertake further study at higher levels of education, f.ex. at the MSc level.
- Familiarize themselves with new knowledge quickly
- Work constructively and with an open mind on in their pursuit of knowledge