HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

SCHOOL OF COMPUTER SCIENCE FALL 2017

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

SCHOOL OF COMPUTER SCIENCE FALL 2017

MSc students can choose courses from either the BSc or MSc routes, but you must get permission from your home university and the department at Reykjavik University. It is not possible for BSc students to take their final thesis project during their exchange studies in the School of Computer Science.

Department Contact:

Please contact the department for information regarding; courses and course selection.

Undergraduate/ BSc/ BA / First cycle
Graduate /MSc/ Masters/ Second Cycle

Undergraduate/BSc
Graduate /MSc
Sigurbjörg Ásta Hreinsdóttir
astahr@ru.is

COURSES OVERVIEW

COURSE CODE	LEVEL	COURSE TITLE	ECTS
T-513-CRNU	BSc	CRYPTOGRAPHY AND NUMBER THEORY	6
T-519-STOR	BSc	THEORY OF COMPUTATION	6
T-603-THYD	BSc	COMPILERS	6
T-409-TSAM	BSc	COMPUTER NETWORKS	6
T-504-ITML	BSc	INTRODUCTION TO MACHINE LEARNING	6
T-511-TGRA	BSc	COMPUTER GRAPHICS	6
E-402-STFO	BSc	MATHEMATICAL PROGRAMMING *3 WEEK COURSE	6
T-809-DATA	MSc	DATAMINING AND MACHINE LEARNING,	8
T-810-OPTI	MSc	OPTIMIZATION METHODS	8
T-811-PROB	MSc	APPLIED PROBABILITY	8
T-719-STO4	MSc	THEORY OF COMPUTATION	8
T-723-VIEN	MSc	VIRTUAL ENVIRONMENTS	8
E-402-STFO	MSc	MATHEMATICAL PROGRAMMING *3 WEEK COURSE	8
T-504-ITML	MSc	INTRODUCTION TO MACHINE LEARNING	6
T-511-TGRA	MSc	COMPUTER GRAPHICS	6
T-603-THYD	MSc	COMPILERS	6
T-740-SPMM	MSc	SOFTWARE PROJECT MANAGEMENT	8
T-814-PROD	MSc	INTEGRATED PRODUCT DEVELOPMENT; CONCEPTS & PROCESSES	8
V-707-BENT	MSc	BECOMING ENTREPRENEUR	7,5
V-720-MINN	MSc	INNOVATION	3,75
V-864-VENT	MSc	VENTURE CAPITAL	7,5

UNDERGRADUATE COURSES:

T-513-CRNU CRYPTOGRAPHY AND NUMBER THEORY — COMPUTER SCIENCE

LEVEL: BSC **ECTS: 6** **PREREQS: Discrete Math, Calculus and Statistics, Algorithms**

CONTENT: We treat the basics of cryptography & number theory. We start with some classical ciphers & the tools from number theory necessary for doing cryptography. We cover symmetric & asymmetric ciphers. Some topics from groups, rings & fields will be introduced & used, especially when we look at elliptic curve cryptography.

LEARNING OUTCOMES: Knowledge:
- Purpose of cryptography and its uses throughout history
- Basics of number and information theory, especially relating to cryptography
- Sage programming language, especially how to implement algorithms from number theory and cryptography
- Common algorithms used in cryptography
- Finite fields, how they are used in cryptography
- Basics of elliptic curves and how they are used in cryptography
- How cryptography is applied, e.g., in multi-party computation, zero knowledge proofs, digital cash and voting systems

Skills:
- Simple cryptographic methods to encrypt short texts by hand
- Write code in Sage use powerful cryptographic methods to encrypt text
- Solve number theoretic problems
- solve problems in other mathematical courses, especially where algebra is needed
- Use Sage as a tool in other programming and mathematical courses, for testing conjectures, drawing graphs, etc.

TEACHING & ASSESSMENT: Assessment Methods: 50% homework (4 best of 5 count) and 50% written final exam.
Textbook: An Introduction to Mathematical Cryptography, By: Hoffstein, Pipher and Silverman ISBN Number: 978-1-4939-1710-5

T-519-STOR THEORY OF COMPUTATION — COMPUTER SCIENCE

LEVEL: BSc **ECTS: 6** **PREREQS: Discrete Math, Algorithms**

CONTENT: Various types of finite automata are introduced and connected to the formal definition of a programming language. Turing machines are introduced as a theoretical model for computation. Computability is discussed and the classification of solvable and unsolvable problems. Finally there is a discussion of complexity classes and the classification of algorithmically hard and easy problems.

LEARNING OUTCOMES:
Knowledge:
- Deterministic & undeterministic Finite Automata (DFA and NFA), regular languages & their most important properties.
- Know what it means that two such automata are equivalent.
- Know the Pumping Lemma for regular languages.

Skills
- draw DFAs & NFAs & describe in words Language they accept.
- Draw an NFA (or DFA) for a simple regular language from the description of that language.
- Describe the strings in a regular language
- Write regular expression that describes a simple regular language based on its description in words.

Competence:
- Use finite automata and their properties in computer science.
- Use the properties of context free grammars in programming.

TEACHING & ASSESSMENT: Home Assignments: 35%, Midterm: 15%, Final exam: 50%

T-603-THYD COMPILERS — COMPUTER SCIENCE

LEVEL: BSC ECTS: 6 **PREREQS: Programming Languages.**

CONTENT: Compilers are the most important part of a programming development environment. The course defines the function & objective of a compiler. Lexical analysis of programs is discussed in detail, regular expression & finite automatons defined and the use of Lex introduced. Top-down and bottom-up approaches in parsing are discussed precisely & the use of Yacc introduced. Implementation of error handling illustrated, particularly semantic analysis. Finally, code generation is covered. Construction of a compiler will be a large component of the course.

LEARNING OUTCOMES: Knowledge
- Understand the structure and design of compilers
- Understand role and function of lexical analyzers, parsers & code generators
- theoretical foundation necessary for compiler construction
- Be able to design and build a simple compiler

Skills
- use regular expressions & finite machines when doing lexical analyzation
- use fragmented grammar and both above- and bottom up parsing methods
- use the software that makes lexical analyzers and parsers

TEACHING & ASSESSMENT: Compiler project (in 3 parts): 37%, Home exercises (3): 18%, Final exam: 45%,

T-409-TSAM COMPUTER NETWORKS — COMPUTER SCIENCE

LEVEL: BSc **ECTS: 6** **PREREQS: Data Structure, Computer Architecture**

CONTENT: We begin with a short overview of network systems and services. Then we will focus will be on the layers of the OSI and IETF models. The following network layers will be studied in the details: application layer (WWW, HTTP, DNS, SMTP, FTP etc), transport layer (UDP and TCP), network layer (link state routing and distance vector routing, IP, IP-addresses, link layer (MAC, Ethernet, Hubs and switches). Finally an introduction to more specific topics such as mobile networks, multimedia networking, and network security.

LEARNING OUTCOMES:
- Explain layers of OSI & IETF network protocol stack & their interactions.
- Explain basic application layer protocols such as HTTP, SMTP, and P2P applications.
- Discuss & analyse the transport layer protocols TCP and UDP.
- Explain details of the IP protocol. Understand link-layer protocols & technology . Explain basic security terminology, security threats in networks, countermeasures, symmetric and public key
- Discuss performance assumptions and scalability in computer systems and networks.cryptography.
- Partition network into subnets according to user specifications.
- Explain and analyse traces of real-world network traffic.
- Work with application layer networking interfaces.
- Understand how to write network services securely.
- Write simple network service using network primitives.
- Write secure network applications.

TEACHING & ASSESSMENT: Homework (24%), programming assignments (30%) and final exams (46%). Textbook: Computer Networking - A Top-Down Approach By: James F Kurose , Keith W Ross ISBN: 9780132856201

T-504-ITML INTRODUCTION TO MACHINE LEARNING — COMPUTER SCIENCE

LEVEL: BSC ECTS: 6 **PREREQS: Algorithms, Calculus and Statistics, Discrete Math**

CONTENT: This course presents an overview of the field of machine learning, which deals with finding patterns and rules in large datasets. Such rules can then be used to predict outcomes of future events, for example with the aim of improving decision making in a wide range of business and manufacturing disciplines. In this course we will study machine learning techniques for classification and clustering as well as other selected techniques. In addition to introducing the underlying theory the the methods will be used to solve practical problems.

LEARNING OUTCOMES: Knowledge:
- Know how data mining is carried out
- Recognize different types of training data and how to deal with common problems that arise, such as incomplete data
- Be familiar with key algorithms and models used for classification, including decision trees, set of rules, Naive Bayes, neural networks and support vector machines
- Basic algorithms used with clustering, including K-means
- Basic algorithms used to find relationships in data (e.g., association analysis)
- Be familiar with basic ideas behind evolutionary and reinforcement learning

Skills:
- Use software tools and programming libraries for data mining to categorize and cluster data
- Be able to set up problems and apply data mining techniques to solve them

Competence:
- Be able to determine the mechanical data learning strategies best suited to the solution of various practical problems, and ready to use data mining tools and libraries to their solution

TEACHING & ASSESSMENT: Homework assignments and in-class quizzes (20 %), Programming assignments (30 %), Final Exam (50%) Textbook: Introduction to Data Mining, By: P, Tan, M Steinback and V Kumar

T-511-TGRA COMPUTER GRAPHICS — COMPUTER SCIENCE

LEVEL: BSc **ECTS: 6** **PREREQS:**

CONTENT: Computer graphics is an increasing part of today's programmer projects. The first part of this course covers the use of the OpenGL library, vector tools for graphics, transformations of objects and polygonal meshes. The second part deals in more detail with three-dimensional drawing with emphasis on perspective, depth, light and colour. In the end, several issues regarding the implementation of a renderer are presented, in addition to curve and surface design. During the course students build several programs related to the course material.

LEARNING OUTCOMES:
Knowledge:
- Be familiar with the algorithms and calculations used when three-dimensional images are drawn on screen in real time (pipeline graphics), including, model transformations, perspective transformations, lighting, shading, clipping and rasterization.

Skills:
Be able to use the OpenGL standard to draw a three-dimensional image on a screen?
- Be able to implement a drawing loop which draws a motion picture, frame by frame, in real time.
- Be able to implement a programming loop that receives input and output, moves things, makes decisions and draws each

- Be familiar with methods in OpenGL that implement these algorithms and calculations and how they are used in graphics applications such as computer games (OpenGL pipeline).
- Know how the flow in a graphical real-time application (i.e. computer game) is implemented, with respect to input, movement and drawing

frame with respect to camera angles and objects in a three-dimensional space.

Competence:

- Be able to implement three-dimensional video games and real-time animations with the OpenGL standard.

TEACHING & ASSESSMENT: Assessment Methods: Programming Assignments: 50%, Problem hand-ins: 10%, Final Exam: 40%. Textbook: Introduction to Computer Graphics By: David J. Eck.

E-402-STFO MATHEMATICAL PROGRAMMING — COMPUTER SCIENCE

LEVEL: BSC ECTS: 6
3 WEEK COURSE

PREREQS: Discrete Mathematics, Calculus and Statistics, Algorithms

CONTENT: Mathematics is generally discovered through experiments. Traditional tools for such experiments are pen and paper, and, of course, the mind. A (historically) recent addition to these tools is the computer. We will look at problems from several areas of mathematics and, in particular, how programming can be used as a means to better understand and ultimately solve those problems. This will involve designing and implementing algorithms, experimentation to make conjectures, and deductive/formal mathematics to prove conjectures. For programming we will use python/sage.

LEARNING OUTCOMES:

Knowledge

- After the course the students should know how computers and algorithms are used in research, both in mathematics and computer science.
- Students recognize linear programming as a method for solving problems
- Students recognize dynamic programming as a method for solving problems
- Students recognize search methods as a method for solving problems
- Students recognize brute force and other common solution methods for solving problems
- Students know several objects from discrete mathematics, such as permutations, graphs, games (like the Game of Life) and finite surfaces (tori and the Klein bottle).

Skills

- Students can use a computer to test conjectures and run simulations.
- Students can use dynamic programming to solve problems
- Students can use linear programming to solve problems
- Students can use search and other common methods to solve problems
- Students can choose an appropriate method to deal with different problems
- Students can prove certain problems by hand, where running simulations is to time-consuming.

Competence

- Students can use the Sage computer algebra system to assist them in other courses
- Students recognize which kind of problems can be solved with the solution methods treated in the course

GRADUATE COURSES

T-809-DATA DATAMINING AND MACHINE LEARNING — COMPUTER SCIENCE

LEVEL: MSC **ECTS: 8** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

CONTENT: Pattern recognition system, classifier design cycle and learning. Statistical pattern recognition, Bayesian decision theory, maximum likelihood and Bayesian parameter estimation. Linear models for classification. Principal component analysis. Multilayer neural networks. Nonparametric methods: k-nearest neighbours and Parzen kernels. Kernel methods and support vector machines. Unsupervised classification, K-means clustering, Gaussian mixture models and expectation maximization. Combination of classifiers, bagging and boosting.

LEARNING OUTCOMES:
Knowledge: After the course the students should be able to recall, describe and define, the following terms: Pattern recognition system, classifier design cycle and learning. Statistical pattern recognition, Bayesian decision theory, maximum likelihood and Bayesian parameter estimation. Linear models for classification. Principal component analysis. Multilayer neural networks. Nonparametric methods: k-nearest neighbours and Parzen kernels. Kernel methods and support vector machines.

Skills: After the course the students should be able to apply the data mining methods and implement the machine learning algorithms presented in the course using standard programming languages such as Python or Matlab and software packages such as scikit-learn and Weka.
Competence: After the course the students should be able to design a suitable machine learning algorithm for a real world problem, evaluate its performance, compare different designs and implementations and interpret the results. The students should also be able to present findings and new results in the subject.

TEACHING & ASSESSMENT: 15% Quizzes 8-10 short computerized quizzes based on material from lectures and readings of which 50% is the outcome of the quiz and 50% is participation in the discussion afterwards. 10% Papers Two overview exams 35% Homework 4-5 papers that include computer exercises, problem solving and academic research 40% Course project One group project of 2-3 people (see detail in hand-out)

T-810-OPTI OPTIMIZATION METHODS — COMPUTER SCIENCE

LEVEL: MSC **ECTS: 8** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

CONTENT: This course introduces the principal algorithms for linear, network, discrete, nonlinear, dynamic optimization and optimal control. Emphasis is on methodology and the underlying mathematical structures. Topics include the simplex method, network flow methods, branch and bound and cutting plane methods for discrete optimization, optimality conditions for nonlinear optimization, interior point methods for convex optimization, Newton´s method, heuristic methods, and dynamic programming and optimal control methods.

T-811-PROB APPLIED PROBABILITY — COMPUTER SCIENCE

LEVEL: MSC **ECTS: 8** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

CONTENT: Overview and approach: This heuristically and practically motivated course will discuss the computation of probabilities of events, discrete/continuous random variables, conditioning of random variables. In addition, the course will also cover transformations of random variables, markov processes, and the applications of stochastic processes to queuing theory, derivatives/finance, decision theory and game theory.

LEARNING OUTCOMES: Understand the basic concepts of probability distribution functions and their role in the modelling of uncertain outcomes – both in the discrete and the continuous case. Use expectation values, variances and covariances to model various probabilistic phenomena

T-719-STO4 THEORY OF COMPUTATION — COMPUTER SCIENCE

LEVEL: MSC **ECTS: 8** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

CONTENT: The main topic of this course is the theoretical basis of computer science. Various types of finite automata are introduced and connected to the formal definition of a programming language. Turing machines are introduced as a theoretical model for computation. Computability is discussed and the classification of solvable and unsolvable problems. Finally there is a discussion of complexity classes and the classification of algorithmically hard and easy problems.

LEARNING OUTCOMES:

Knowledge:

- Deterministic and undeterministic Finite Automata (DFA and NFA), regular languages and their most important properties.
- Correspondence between finite automata and regular expressions.
- Know the Pumping Lemma for regular languages.
- Know context free grammars, context free languages and push-down automata and the correspondence between the concepts.
- Know Turing machines and different variants of those.

Skills:

- Describe the strings in a regular language from a regular expression that describes it.
- Write a regular expression that describes a simple regular language based on its description in words.
- Draw a Turing machine and describe the language it accepts.
- Show that a language is decidable by using closure properties for such languages.
- Decide if a language belongs to the complexity class P or NP.

Competence:

- Finite automata and their properties in computer science.
- Reason about how difficult problems are to solve according to their possible decidability and their complexity.

TEACHING & ASSESSMENT: There will be five home assignments, each worth 7%. Midterm: 15% Final exam: 50%. Both the midterm and the final exam will be closed book. For the MSc students, the final exam will consist of two parts. The final written exam will weigh 45% of the final grade. Moreover, there will be an oral exam on Rice´s theorem that will weigh 5% of the final grade. The oral exam will consist of a presentation, lasting at most 15 minutes including questions, using the whiteboard.

T-723-VIEN VIRTUAL ENVIRONMENTS — COMPUTER SCIENCE

LEVEL: MSC **ECTS: 8** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

CONTENT: This is a comprehensive course in both the theory and practice of Virtual Environments (VEs). Virtual Environments are simulations that engage the senses of users through real-time 3D graphics, audio and interaction to create an experience of presence within an artificial world. VEs are used in a variety of settings, including training, education, health, online collaboration, scientific visualization and entertainment. Their use is becoming more and more pervasive as hardware gets more capable of simulating reality in real-time (including graphics, physics and intelligent behavior). As part of the theoretical overview, the course will introduce the history of VEs, what kind of problems VEs have proven to be best at addressing, what are their shown limitations, what models of human-computer interaction apply to VEs and how these models are evolving and pushing the state-of-the-art in interactivity. The technical portion of the course will lead students through the construction and population of VEs in a very hands-on manner, covering topics such as world representation, real-time graphics and simulation issues, networked environments, avatars and interactive characters, event scripting and AI control, special real-time visual and aural effects and intuitive user interfaces.

LEARNING OUTCOMES:

- Know what constitutes a virtual environment, why they have been created throughout history and how they are used today.
- Be able to think critically about virtual environments as a user interface and design effective environments.
- Understand how humans construct a mental image of their environment using visual cues and how this can be exploited.
- Know the difference between presence and immersion, and understand how these may be measured.
- Understand the principles of effective action in virtual environments, including concepts such as flow, implicit constraints, explicit constraints and contextual action.
- Be familiar with the roles of characters in virtual environments and the common ways to make them autonomous and to animate them.
- Know what an avatar is and understand the issues that relate to level of control.
- Be familiar with the several techniques for constructing visual realism in virtual environments.
- Be able to create an interactive virtual environment in a scripting language and use a scene representation, models, terrain, lights, texturing, physics, animation, heads-up-display and shaders.

TEACHING & ASSESSMENT: Programming Assignments (x2) 20%, Final Project Proposal 5%, Final Programming Project 30%, Final Project Report 5%, Discussion Prep and Lab Work 10%, Final Written Exam 30%,

E-402-STFO MATHEMATICAL PROGRAMMING — COMPUTER SCIENCE

LEVEL: MSC ECTS: 8 | **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

CONTENT:. Mathematics is generally discovered through experiments. Traditional tools for such experiments are pen and paper, and, of course, the mind. A (historically) recent addition to these tools is the computer. We will look at problems from several areas of mathematics and, in particular, how programming can be used as a means to better understand and ultimately solve those problems. This will involve designing and implementing algorithms, experimentation to make conjectures, and deductive/formal mathematics to prove conjectures. For programming we will use python/sage.

LEARNING OUTCOMES:

Knowledge

- Know how computers and algorithms are used in research, both in mathematics and computer science.
- recognize linear programming as a method for solving problems
- recognize dynamic programming as a method for solving
- recognize search methods as a method for solving problems
- recognize brute force and other common solution methods for solving problems
- know several objects from discrete mathematics, such as permutations, graphs, games (like the Game of Life) and finite surfaces (tori and the Klein bottle).

Skills

- can use a computer to test conjectures and run simulations. can use dynamic programming to solve problems
- can use linear programming to solve problems
- can use search and other common methods to solve problems
- can choose an appropriate method to deal with different problems
- can prove certain problems by hand, where running simulations is to time-consuming.

Competence

- Use the Sage computer algebra system to assist them in other courses

T-504-ITML INTRODUCTION TO MACHINE LEARNING — COMPUTER SCIENCE

LEVEL: MSC ECTS: 6 | **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

CONTENT: Overview of the field of machine learning, which deals with finding patterns and rules in large datasets. Such rules can then be used to predict outcomes of future events, for example with the aim of improving decision making in a wide range of business and manufacturing disciplines. We will study machine learning techniques for classification, clustering, and association analysis as well as other selected techniques. In addition to introducing the underlying theory the methods will be used to solve practical problems.

LEARNING OUTCOMES: Knowledge: Know how data mining is carried out. Recognize different types of training data and how to deal with common problems that arise, such as incomplete data. Be familiar with key algorithms and models used for classification, including decision trees, set of rules, Naive Bayes, neural networks and support vector machines. Know the basic algorithms used with clustering, including K-means. Know the basic algorithms used to find relationships in data (e.g., association analysis). Be familiar with basic ideas behind evolutionary and reinforcement learning. Skills Be able to use software tools and programming libraries for data mining to categorize and cluster data. Be able to set up problems and apply data mining techniques to solve them. Competence Be able to determine the mechanical data mining strategies best suited to the solution of various practical problems, and be ready to use data mining tools and libraries to their solution.

T-511-TGRA COMPUTER GRAPHICS — COMPUTER SCIENCE

LEVEL: MSC ECTS: 6 | **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

Computer graphics is an increasing part of the projects of today´s programmer. The first part of this course covers the use of the OpenGL library, vector tools for graphics, transformations of objects and polygonal meshes. The second part deals in more detail with three-dimensional drawing with emphasis on perspective, depth, light and color. Finally, several issues regarding the implementation of a renderer are presented, in addition to curve and surface design. During the course students build several programs related to the course material.

Knowledge: Be familiar with the algorithms and calculations used when three-dimensional images are drawn on screen in real time (pipeline graphics), including, model transformations, perspective transformations, lighting, shading, clipping and rasterization. Be familiar with methods in OpenGL that implement these algorithms and calculations and how they are used in graphics applications such as computer games (OpenGL pipeline). Know how the flow in a graphical real-time application (i.e. computer game) is implemented, with respect to input, movement and drawing Skills: Be able to use the OpenGL standard to draw a three-dimensional image on a screen. Be able to implement a drawing loop which draws a motion picture, frame by frame, in real time. Be able to implement a programming loop that receives input and output, moves things, makes decisions and draws each frame with respect to camera angles and objects in a three-dimensional space. Competence: Be able to implement three-dimensional video games and real-time animations with the OpenGL standard.

T-603-THYD COMPILERS

LEVEL: MSC **ECTS: 6** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

Content

Compilers are the most important part of a programming development environment. The course defines the function and objective of a compiler. Lexical analysis of programs is discussed in detail, regular expression and finite automatons defined and the use of Lex introduced. Top-down and bottom-up approaches in parsing are discussed precisely and the use of Yacc introduced. Implementation of error handling illustrated, particularly semantic analysis. Finally, code generation is covered. Construction of a compiler will be a large component of the course.

Learning outcome - Objectives

Knowledge •Understand the structure and design of compilers •Understand the role and function of lexical analyzers, parsers and code generators •Be able to get a theoretical foundation necessary for compiler construction Skills •Be able to use regular expressions and finite machines when doing lexical analyzation •Be able to use fragmented grammar and both above- and bottom up parsing methods •Be able to use the software that makes lexical analyzers and parsers Competence •Be able to design and build a simple compiler

T-740-SPMM Software Project Management

LEVEL: MSC **ECTS: 8** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

TBA

T-814-PROD Integrated Product Development; Concepts & Processes

LEVEL: MSC **ECTS: 8** **PREREQS: BSC DEGREE – COMPUTER SCIENCE**

Content

We cover the engineering approach to innovation and entrepreneurship in lectures and a practical program in an active company.
Due to increasing freedom in trade and internationalization the competition between companies is boosting. At the same time consumers demand new solutions, and the technology develops, resulting in older solutions becoming obsolete. Such conditions require constant innovation in companies management and an understanding of the nature of innovation and entrepreneurship.
Innovation is not only necessary in technological companies, but in all companies that intend to live and prosper.
The course will cover innovation and the ability companies have for innovation in light of market, science, engineering, planning and financial presumptions. We deal with the terms innovation and entrepreneurship and their significance for modern management and put in context with success. We will also cover the value of knowledge, intellectual property rights and patent rights. Then we cover the internationalization and its impact on the innovation process.
Special emphasis will be put on systematic development of the processes connected to innovation and worked on a project in a real company in this field.

Learning outcome - Objectives

T 814 PROD - Learning outcomes: At the end of the course the students shall have reliable knowledge of the methods used creating innovation basis and be capable to develop and construct a system for managing innovation in companies. Knoweldge: · Understand the presumption for success and the reasons for mistakes in innovation within companies. · Understand how companies can develop, maintain and increase their skill for innovation and the value of innovation and initiative thinking for the existence of companies. · Knowing companies methodology for developing products and innovation and pioneer thinking and the development of new products (goods and service). Skills: · Be familiar with companies methodology for developing products and innovation and being able to use it. · Posses good knowledge of the main items of the innovator science and adaptation and integration of the knowledge of individual employees in order to create strong teams. · Be able to evaluate the reasons for success and evade mistakes in innovation within companies. Competence: · Can by themselves take on a systematically construction and the processes connected to innovation in companies and possess the understanding, skill and knowledge to manage the development and running of such systems within companies. · Be able to introduce and interpret the conclusions and proposals on the above mentioned fields and be able to express themselves on those issues.

Course assessment Reports (4), each 18% total 72%. Verbal exam 28%.

V-707-BENT BECOMING AN ENTREPRENEUR

LEVEL: MSC ECTS: 7,5 | **PREREQS: NONE**

ContentTo act and think in entrepreneurial and innovative ways is an essential competency in post-modern societies, characterised by continuous change and ruptures. Entrepreneurs can be defined, not only as those capable of adapting to change, but those able to act as a force of change. It is a well-supported claim that no longer can we rely on learned expertise to ensure future careers. Acting on this challenge, the aim of this course is to introduce and involve students to adopt entrepreneurial thinking and skills. We will move away from the common 'heroic' idea of the born entrepreneur, but instead entrepreneurship is studied as a set of practices, involving how we create and approach new challenges, identify and act on opportunities. These have become important skills in the wider social context, not only for people aspiring to start their own businesses, but also for those wanting to increase their ability to approach complex, even unclear, challenges and solve them in innovative and creative ways. The course addresses entrepreneurship as force of change and novelty in two combined steps:

The course takes a practice- based approach to enhance students´ entrepreneurial competencies, by exercising essential skills like creativity, empathy, and collaboration, and by working with critical phases of the entrepreneurial and start-up process (e.g. idea generation – qualification – feedback – execution). Introducing entrepreneurship as a field of research-based knowledge, which offers valuable insights to, for example, entrepreneurial processes; what evokes entrepreneurship; risk and success factors. This knowledge adds to a foundation to guide entrepreneurial decision-making and action.Classes are a combination of lectures and exercises, relying on students' active participation. In sum, the overarching question we want to tackle is: How do we become entrepreneurial in thought and practice?

V-720-MINN INNOVATION

LEVEL: MSC ECTS: 3,75 | **PREREQS: NONE**

Content

Innovation has become a 'mantra' of post-industrial organisations. In a 'globalized', 'digitalized', and 'mediatized' world changes are perpetual and swift. Innovation is an essential and demanded capacity of successful organizations, managers, and employees. This course addresses innovation in an organisational context, emphasizing central aspects and skills constituting innovative processes, including creating, identifying, evaluating, and exploiting good ideas. Students will be introduced to different approaches to innovation (e.g. open innovation and design-driven innovation); innovation in different industries; and from different perspectives (e.g. management, service/product development, and marketing). The course is a practical and theoretical preparation for specific situations the students will find and have to manage after having finished their studies. Classes are a combination of lectures, cases-studies and workshops. There is emphasis on active participation of students and engagement with innovative practices.

Learning outcome - Objectives

Knowledge: Collection of facts, concepts, theories and techniques acquired by students. The student should: Understand the importance of innovation for sustainability and growth of organisations. Understanding how innovation is a challenge for established organisations. Be able to account for different approaches to innovation and organisational entrepreneurship. Skills: Ability to apply knowledge to different tasks of innovative processes. The student should: Be able to critically assess different approaches to organisational innovation. Be able to develop a plan to absorb, process, and execute innovative ideas in/for organisations. Be able to identify differences between traditional organisational management and leading innovations in organisations. Competences: Ability to apply knowledge and skills to doing innovation. The student should: Be able to lead a process towards discovering and executing innovative ideas. Be able to develop new and innovative ideas.

Teaching and learning activities

Teaching will be a combination of different approaches e.g. lectures from teacher, practice-based teaching, group-work, and student presentations. The teaching relies on the students' active participation and good preparation. The teaching methods are designed to reflect the stated learning outcomes of the course.

V-864-VENT VENTURE CAPITAL

LEVEL: MSC **ECTS: 7,5** **PREREQS: NONE**

Content

Venture capital refers to equity investments made for the launch, early development, or expansion of a business. It is therefore directly relevant for the development of new products and technologies and the early growth of innovative ventures. Venture capital funds perform an important intermediary function: they channel funds from institutional investors to realize and capture value from the development of high-potential ventures or from improving the operational efficiency of existing companies. The purpose of this course is to expose students to the concepts and operation of venture capital funds and develop their knowledge, skills, and competences associated with the four stages of the venture capital cycle: fundraising, investing, value adding, and exiting.

Learning outcome - Objectives

Knowledge: Collection of facts, concepts, theories and techniques acquired by students. The student should: Identify the different stages of and relationships in the venture capital cycle. Describe the operation of venture capital firms / funds. Explain the key issues associated with fundraising, investing, value adding, and exiting from the point of view of venture capital managers. Skills: The ability to apply knowledge to different tasks of the venture capital process. The student should: Analyze the issues relevant in venture capital investment situations. Categorize the risks associated with venture capital transactions. Compare different fundraising or investment options. Competences: The ability to apply knowledge and skills in venture capital settings. The student should: Evaluate investment proposals. Formulate due diligence plans. Design funding instruments that align the incentives of different parties to a venture capital transaction.

This is an intensive course, conducted over 6 days, in two 3-day blocks. The time spent on the course will be spread roughly equally among interactive lectures, group work / preparation, and class discussion. Interactive lectures will be used to introduce the major topics of the course and provide the necessary background for case discussions and class exercises. Students are expected to read all the assigned material beforehand and to present / discuss it in class. Most of the real learning will occur through analysis and discussion of these real-world cases and exercises, both in class and (more importantly) in the student's own preparation, group interaction, and subsequent reflection on the material.