Studieplan 2015/2016

Bachelor in Game Programming

Studyprogramcode
BSP
Short description
Game programming is a broad field of study where the skills required are relevant to many industries. Game programmers must primarily be good programmers with a broad understanding of computer programming and technology and the ability to work with a very wide range of other disciplines. The core skills required range from understanding real-time graphics techniques to multiplayer networking and computer operating systems. Computer games have driven the development of new technology and push current computing devices to their limit. Computer game programmers have to understand how to get the most out of limited hardware, such as mobile phones, to create enjoyable experiences.

This degree is not an easy option. This degree requires all the programming courses in a standard programming degree, and includes course on graphics, game programming, artificial intelligence, mobile devices, and parallel programming. Graduates of the course are expected to be both good programmers and have the ability to communicate with designers and artists. The degree requires participation in many group projects and presentations.

After graduation, you will have a solid education in programming that allows access to jobs in gaming companies, software engineering, software development and the entertainment industry. The program also provides a good basis for working in: Systems for e-learning, visualization, simulation, interactive web applications and graphics-based computer systems.

Many of the courses are taught English and have an international focus. The curriculum is based on the International Game Developers Association's curriculum framework, the ACM's computer science curriculum, and the UK SkillSet game industry accreditation guidelines. This international focus allows graduates to work anywhere in the world.

Duration
The program is a 3-year undergraduate computer education that provides 180 study points spread over six semesters. Upon completion you are awarded the degree of a Bachelor in Game Programming. This program is eligible for admission to IT and CS related master's programs in Norway and abroad.

Expected learning outcomes
After completing the study you will have the ability to develop computer games from scratch or using a game engine. Our graduates are able to work as part of a large team to develop interactive applications and computer games. In particular, you will have the ability to implement advanced graphics, artificial intelligence, networking and interaction.

The specific learning outcomes include:

Knowledge

- A basic knowledge of maths, algorithms and problem solving.

- Knowledge of the communication and information technology used for computer games.
- Knowledge of Computer Graphics, Artificial Intelligence, and mobile systems as core elements in computer game technology.
- An understanding of the business requirements for making computer games.
- Can explain the need for professional work methods for development of computer games.
- An understanding of the legal, social, ethical consequences of computer technology and computer games.

Skills

- The ability to apply what you have learned to solve practical problems in computer programming and game development.
- Collaboration in teams to develop and present solutions to problems both orally and in written form.
- The ability to find relevant sources in text and on the Internet to solve algorithmic, design, and technical problems.
- An understanding of the role of middleware and tools and the ability to use these tools in conjunction with personally developed code.
- The candidate can use tools which support development of computer games.

General Competence

- In insight into academic forms of communication and the ability to write reports in English.
- The ability to develop innovative solutions to tasks, with a focus on rapid prototyping.
- An understanding of entrepreneurial activity and the ability to be part of startup businesses.
- The ability to update your own knowledge and continue life-long learning.

Along with these game programming specific abilities our graduates are expected to be good citizens. To understand the role of computers and games in modern society. Our graduates should contribute to the debate about the role of games in society and be able to make ethical decisions about the nature and content of the games and software they help create.

In your final semester you will develop a game as part of the Bachelor's thesis. This project will form a critical part of your portfolio and CV. The project will also have input from professional game developers, and the results will be reviewed by industry partners. This capstone project will help integrate all the subjects that are part of the bachelor degree and provide vital group work experience.

Target Group
There is no requirement for particular computer skills before the course starts. The program assumes that students are interested in computer games and fascinated with how they are made. This course is designed to engage any student who meets the admission requirements and are motivated to immerse themselves in this exciting and competitive field.

Students from high school are the primary target group, but the program is also suitable for those who are returning from the workforce looking to develop there programming and game development ability.

Admission Criteria
Admission to the program is general admission plus R1/S2 (2MX) or real competence. If you do not have the required mathematics background, you can apply under the condition that you take the summer courses (R1) in mathematics offered by the university college, which starts a few weeks before

the ordinary studies.

Course Structure
As a student in the Bachelor's degree in game programming, you will gain general computer science skills, which characterizes any skilled programmer. It is important to have a comprehensive understanding of the computer environment a game works within, from understanding the hardware and data networks to end users and game experience. The program is built up with elements of both basic computer science courses and games specific topics. The last part of the program heavily specialises in games programming.

The program will therefore have a particular focus on:

- **Programming:** Basics and language-independent skills in structured and object-oriented programming.
- **Graphics:** Theory and techniques in 2D and 3D graphics.
- **AI:** Understanding how computers control agents in virtual worlds.
- **Mobile Devices:** Mobile devices are the fastest growing sector of game development
- **Parallel Programming:** Utilizing the full power of modern CPUs and GPUs

Many courses use projects to embed theory in practical implementation. The program ends with a major game development project (bachelor thesis) which can be in association with industry or can be used as the basis for forming a new company. The college offers modern laboratories tailored to the needs of the individual topics. You will engage with lecturers from around the world in this highly skilled professional community. In addition, you will be involved in research activities under the direction of the department's professors. You will experience an inspiring and active study environment with students from various IT and media programs.

Some topics are shared with Computer Science and Engineering. This includes topics such as Operating Systems and Data Communications, which provides understanding of the underlying technological platform of different systems and the Internet. Programming skills are built up throughout the study in the topics Basic programming, Object-oriented programming and algorithmic methods courses. Other topics are game specific including Game Design, Mathematics for Game Programming, Graphics Programming, Game Programming and Artificial Intelligence.

20 credits are elective courses. Students will also be able to choose among a wide range of science topics from other IMT studies.

Internal/external examiner
External sensors are used in many parts of the program, depending on the assessment form in the respective topics. The external sensors are used both for grading of exam papers, as well as reviewing the overall academic level and content of the degree.

Internationalization
Students can go abroad in semester 4 provided that they find a study program containing Operating Systems and Software Engineering. Contact the international office at GUC student administration for specific information, help and advice. As many of the courses are taught in English the program is also very well suited to continuing your studies at the Masters level aboard.

Publiser

Yes
Degree
Bachelorgrad

Bachelor of Game Programming, 1st year 2014/2015

Coursecode	Course name	C/E *)	ECTS each. semester					
			S1(A)	S2(S)	S3(A)	S4(S)	S5(A)	S6(S)
IMT1361	Game Design	C	10					
REA1101	Mathematics for computer science	C	10					
IMT1031	Fundamental Programming	C	10					
IMT1082	Object-Oriented Programming	C		10				
IMT2431	Data Communication and Network Security	C		10				
REA2061	Mathematics for Game Programming	C		10				
		Sum:	30	30	0	0	0	0

*) C - Compulsory course, E - Elective course

Bachelor of Game Programming, 2nd year 2015/2016

Coursecode	Course name	C/E *)	ECTS each. semester					
			S1(A)	S2(S)	S3(A)	S4(S)	S5(A)	S6(S)
IMT2021	Algorithmic Methods	C			10			
IMT2531	Graphics Programming	C			10			
IMT2571	Data Modelling and Database Systems	C			10			
IMT2581	Rapid Prototyping and Innovation	C			2,5	2,5		
IMT2243	Software Engineering	C				10		
IMT2282	Operating Systems	C				10		
IMT3591	Artificial Intelligence	C				10		
		Sum:	0	0	32.5	32.5	0	0

*) C - Compulsory course, E - Elective course

Bachelor of Game Programming, 3rd year 2016/2017

Coursecode	Course name	C/E *)	ECTS each. semester					
			S1(A)	S2(S)	S3(A)	S4(S)	S5(A)	S6(S)
IMT3601	Game Programming	C					10	
IMT3662	Mobile Development Theory	C					5	
	Elective course, 10 ECTS	E					10	
	Elective course, 10 ECTS	E						10
IMT3912	Bachelor's thesis	C						20
		Sum:	0	0	0	0	25	30

*) C - Compulsory course, E - Elective course

Electives

Coursecode	Course name	C/E *)	ECTS each. semester	
			S1(A)	S2(S)
IMT3102	Object-Oriented Software Development	E	10	
IMT3281	Software Development	E	10	
IMT3861	Stormaskiner	E	10	
IMT3672	Mobile Development Project	E	5	
IMT3801	Multi-threaded Programming	E	5	
IMT2291	Web Technology	E		10
IMT3511	Discrete Mathematics	E		10
IMT3612	GPU Programming	E		5
IMT3602	Professional Programming	E		5
		Sum:	0	0

*) C - Compulsory course, E - Elective course

Emneoversikt

IMT1361 Game Design - 2015-2016

Course code:
IMT1361

Course name:
Game Design

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
English

Expected learning outcomes:
On completion of this course the students will be able to:

Knowledge:

- Understand game mechanics and their interaction
- Understand the process of designing a game

Skills:

- Discuss the game design process using correct terminology
- The ability to design a game based on a theme
- Work through the development of an idea from game concept to rules and implementation

General competence

- Work more effectively in groups
- Understand the basic methods related to innovation and entreprenuership

Topic(s):
The topics covered are varied but will include

- Game design process
- Definitions of games, puzzles, toys and play
- Player motivation
- Game mechanics
- Game balance
- Story and character design
- Interface design
- Creativity in teams
- Documentation

Teaching Methods:
Lectures
Exercises

Teaching Methods (additional text):
This course will draw on the students experience with games and develop an appreciation for the design process by requiring the students to design innovative games within constraints.

Form(s) of Assessment:
Exercises
Written exam, 4 hours

Form(s) of Assessment (additional text):

- Written exam, 4 hours (counts 40%)
- Four assignments (counts 60%)

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Internal examiner, together with external examiner once every three years, next time in 2016.

Re-sit examination:
Re-sit August 2016 for the written exam

Tillatte hjelpemidler:

Coursework Requirements:
The students will cover the content of the 3IKK course during tutorials

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Associate Professor Simon McCallum

Teaching Materials:
Jesse Schell, The Art of Game Design: A book of lenses: Second Edition (2014)

(recommended) Katie Salen and Eric Zimmerman, Rules of Play, Game Design Fundamentals (2004)

Publish:
Yes

Home page:
Additional course information

REA1101 Mathematics for computer science - 2015-2016

Course code:
REA1101

Course name:
Mathematics for computer science

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
Norwegian

Prerequisite(s):
2MX or REA 3022 Matematikk R1 or equivalent course.

Expected learning outcomes:
The students will learn mathematical tools and methods for engineering problem solving, and have a foundation for further study in mathematics and computer science. The course emphasizes applications.

Knowledge

- Understand the relevance of mathematics in engineering problem solving.
- Able to identify applications of mathematics in engineering subjects.
- Know the possibilities and limitations of mathematical software.

The students will have knowledge in the areas of logic and discrete mathematics, with

Skills

- able to understand and use mathematical language.
- able to use mathematical methods and software to solve problems.
- basic mathematical reasoning.

Topic(s):

- Number theory
- Matrices
- Propositional and predicate logic
- Proofs
- Sets, functions and relations
- Enumerative combinatorics
- Graphs and trees
- Automata and languages

Teaching Methods:
Lectures
Mandatory assignments
Exercises

Form(s) of Assessment:
Portfolio Assessment
Written exam, 4 hours

Form(s) of Assessment (additional text):

- Written exam – 4 hours (60%)
- Portfolio (40%)
- The students must pass both the exam and the portfolio.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

Re-sit examination:
Re-sit August 2016 for the Written examination.

Tillatte hjelpemidler:
D: Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Academic responsibility:
Faculty of Technology, Economy and Management

Course responsibility:
Førsteamanuensis Bernt Tore Jensen

Teaching Materials:
Richard Johnsonbaugh: Discrete Mathematics, 7th ed., Pearson/Prentice Hall

Additional material published on classfronter.

Publish:
Yes

IMT1031 Fundamental Programming - 2015-2016

Course code:
IMT1031

Course name:
Fundamental Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
Norwegian

Expected learning outcomes:
Knowledge:

- Read and explain fundamental C++ syntax.
- Analyze the problem for simple programming tasks.
- Find and write the program code for solving such a problem.
- Obtain a suitable/appropriate data structures for a computer program, primarily containing arrays/tables.

Skills:

- Using a program develoment tool containing a C++ compiler.
- Understand and use fundamental C++ syntax.
- Writing program code that is implementation/realization of a self-found or already known algorithm.
- Getting to known and change/modify/expand already existing program code.
- Create and manage simple data structures consisting of arrays/tables.

General Competence:

- Work systematically, structured and targeted to solve a (programming) problem.
- Practical own efforts ("hands on ") is the way to new knowledge and skill.

Topic(s):
Construction of programs:
- Step by step
- Algorithms
- Pseudo code
Introduction to language elements as:
- Program structure and expressions
- Types of data, variables, strings and constants
- Operators
- Flow of control (decisions and loops)
- Structures
- Functions and parameters
- Arrays
- Classes and objects
Use of library functions:
- Streams (files and I/O)
- String handling

Teaching Methods:
Lectures
Mandatory assignments
Exercises

Form(s) of Assessment:
Written exam, 4 hours

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Graded by course instructor and examiner.

Re-sit examination:
Re-sit August 2016

Tillatte hjelpemidler:

Examination support:
All printed matters and hand written notes

Coursework Requirements:
4 of 5 mandatory assignments must be approved by student assistant. No. 1 must be one of them. Clearly inadequate work, not independently own work or deadline that is not complied is considered as undelivered.

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Høgskolelektor Frode Haug

Teaching Materials:
Lafore, Robert. (2002). Object-Oriented Programming in C++. Indianapolis, IN: SAMS.
Faglærer. Kompendium. Gjøvik: HiG.

Additional information:
The course overlaps entirely with IMT1241 Basic programming in Java

Publish:
Yes

IMT1082 Object-Oriented Programming - 2015-2016

Course code:
IMT1082

Course name:
Object-Oriented Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Vår

Language of instruction:
Norwegian

On the basis of:
IMT1031 - Fundamental Programming

Expected learning outcomes:
Knowledge:

- Read and explain more advanced C++ syntax.
- Explain and use object-oriented approach / thinking.
- Find an suitable/appropriate data structure for moderate big computer program.
- Explain the use of a small programming library (toolbox).
- Develop an application (as project work) consisting of a number of different files.
- Understanding the quality aspects of development and maintenance of software.

Skills:

- Understand and use more advanced C++ syntax.
- Solve programming tasks with object-orientation approach/thinking.
- Using and mastering a programming library.
- Choose, create and manage more sophisticated data structures, primarily consisting of lists and arrays/tables.
- Master tools for version control, code analysis and testing.

General Competence:

- Cooperate with other people in a project.
- Analyze, plan and implement a larger work (project).
- Dealing with and adhere to deadlines.

Topic(s):
Principles for object-orientation
Introduction to language elements as:
- Classes and objects (repetition)
- Overloading
- Inheritance
- Pointers
- Dynamic allocation
- Lists
- Virtual functions and late binding

Bigger programs (application) consisting of multi-files.
Tools for version control, code analysis and testing.

Teaching Methods:
Lectures
Mandatory assignments
Exercises
Project work

Form(s) of Assessment:
Written exam, 4 hours

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Graded by course instructor and examiner.

Re-sit examination:
Re-sit August 2016

Tillatte hjelpemidler:

Examination support:
All printed and hand-written support material is allowed.

Coursework Requirements:
2 of 3 mandatory assignments and project work must be approved by student assistant. Clearly inadequate work, not independently own work or deadline that is not complied is considered as undelivered.

The mandatory assignments must be submitted before the student can join a group and start the project work. It requires active participation in the project to get it approved. Group participants must sign a paper dealing that all students have been active/participating, and each one can be extracted for an oral exam to get the project approved.

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Høgskolelektor Frode Haug

Teaching Materials:
Lafore, Robert. (2002). Object-Oriented Programming in C++. Indianapolis, IN: SAMS
Faglærer. Kompendium. Gjøvik: HiG

Publish:
Yes

IMT2431 Data Communication and Network Security - 2015-2016

Course code:
IMT2431

Course name:
Data Communication and Network Security

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Vår

Language of instruction:
English

Expected learning outcomes:
Knowledge:

- The candidate possesses thorough knowledge of models and protocols in data communication networks.
- The candidate possesses thorough knowledge in the theory of network security.
- The candidate is capable of applying his/her knowledge in the field of IT-security.
- The candidate is capable of updating his/her own knowledge in data communication networks.

Skills:

- The candidate is capable of performing basic network administration tasks.
- The candidate is capable of performing error tracking and solving in basic networks.
- The candidate knows relevant methods and terminology in the area of data communications.

General competence:

- The candidate is capable of working independently and in groups in the field of data communication.
- The candidate is capable of designing, analyzing, and performing maintenance on basic networks.

Objectives:
After completion of the course, the students:

- Will have knowledge of the most used standards and protocols for data communication.
- Will understand the principles of network security.

Topic(s):

- Basics of computer networks
- Application layer (HTTP, SMTP, DNS)
- Transport layer (TCP, UDP)
- Network layer (IP, ICMP, routing)
- IPv6 Network addressing
- Data link and physical layer (Ethernet, MAC, ARP, witching, VLANs)
- Basics of network security, including applied cryptography
- Authentication in networks (Kerberos)
- Firewalls
- Network Intrusion Detection Systems

Teaching Methods:
Lectures
Laboratory work
Exercises

Form(s) of Assessment:
Other

Form(s) of Assessment (additional text):
The exam consists of three/(four optional) parts. These include the final exam of CCNA R&S Module 1 + 2, a written subnetting exercise, a router configuration part using Packet Tracer simulation software (optional) and a written exam of network security.
Part 1: - Final Exam (50 points) CCNA Module 1, I2N
Part 2: - Final Exam (50 points) CCNA Module 2 R&S
Part 3: - written exam network security (50 points),
- written exam subnetting (50 points / optional 25 points)
- optional, skill test Packet Tracer (optional 25 points),
All sub-parts must be passed (>=40%) to pass the course
Conversion from 200 point scale to A-F scale according to recommended conversion table. In specific circumstances, emneansvarlig can slightly adjust the limits in the conversion table to enforce compatibility with the qualitative descriptions on the A-F scale.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Evaluated by internal examiner, external examiner is used periodically (every four years, next time in 2015/2016)

Re-sit examination:
Single Parts of the exam can be re-taken again.

Tillatte hjelpemidler:

Examination support:
None.

Coursework Requirements:
None

Academic responsibility:
Faculty of Computer Science and Media Technology

Emneansvarlig kobling:
Thomas Kemmerich

Course responsibility:
Førsteamanuensis Thomas Kemmerich

Teaching Materials:
CISCO Netacademy teaching materials.

Handout articles.

Kurose, J. and Ross, K.W. (2008): Computer Networking: A Top-Down Approach, sixth edition. Addison- Wesley (recommended background material).
William Stallings: Cryptography and Network Security: Principles and Practice 6 Ed. (recommended background material).

Replacement course for:
IMT3371

Publish:
Yes

REA2061 Mathematics for Game Programming - 2015-2016

Course code:
REA2061

Course name:
Mathematics for Game Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Vår

Language of instruction:
English

On the basis of:
REA1101 Mathematics for computer science

Expected learning outcomes:
After completing the subject the students should have knowledge of:

- Applications of mathematical logic
- Algorithms for pathfinding
- The mathematics of 3d-graphics
- Elementary mechanics for games
- The role of mathematics in game development

Skills in:

- Creating matrices for transformations in three dimensions
- Solving the motion interception tasks
- Creating path finding solutions
- Analysing games using probability theory
- Turning mathematical descriptions into programming code

General competence of:

- Improved general logical and mathematical reasoning
- Problems solving and rigorous descriptions of solutions
- General programming ability
- Written and spoken English

Topic(s):

- Logic
  - Logic puzzles
  - Introduction to logic programming languages
  - Parametric logic
  - Bitwise logic
- Probability
  - Elementary probability and enumeration
  - Conditional probability
  - Analysis of games using Markov chains
  - Expectation
- Pathfinding using the A* algorithms
- Three dimensional geometry of 3D graphics
  - Transformations, homogeneous coordinates.
  - Transformations in 3D graphics
  - Complex numbers and quaternions
  - Interpolation
  - Parameterisation of curves and surfaces
  - Ray tracing
- Mechanics
  - Trajectories and equations of motion
  - Elastic and inelastic collisions, collision detection
  - Inverse Kinematics

Teaching Methods:
Lectures
Mandatory assignments
Exercises

Form(s) of Assessment:
Written exam, 5 hours

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Internal examiner, together with external examiner once every three years, next time in 2016.

Re-sit examination:
Re-sit August 2016

Tillatte hjelpemidler:
A: Alle trykte og håndskrevne hjelpemidler tillatt. Alle kalkulatorer tillatt.

Coursework Requirements:
Up to 5 Compulsory assignments

Academic responsibility:
Faculty of Computer Science and Media Technology

Emneansvarlig kobling:
Bernt Tore Jensen

Course responsibility:
Førsteamanuensis Bernt Tore Jensen

Teaching Materials:
Handouts and internet resources.

Publish:
Yes

IMT2021 Algorithmic Methods - Study plans 2016-2017

Course code:
IMT2021

Course name:
Algorithmic Methods

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
Norwegian

On the basis of:
IMT1082 - Object-Oriented Programming

REA1101 - Mathematics for computer science
or
REA2091 - Mathematics 2 for computer science

Expected learning outcomes:
Knowledge:

- Become familiar with, explain, apply and to some extent be able to rewrite some standard algorithms such as sorting, searching and graph handling.
- Describe and explain various data structures (arrays/tables, linked lists, queues, stacks, trees and graphs).
- Analyze advanced and complex (non-trivial) issues, and finding the algorithm to solve these.
- Apply recursive approach/method of problem solving and programming.
- Using abstraction in the construction of programs.

Skills:

- Writing reliable and efficient / fast computer programs.
- Write the program code that addresses advanced and complicated issues.
- Manage and handle advanced data structures (with particular emphasis on trees and graphs).

General competence:

- Had developed the ability to think and solve sophisticated and complex problems.
- Finding other/newer knowledge (here: algorithms), results and research in the field.

Topic(s):
Techniques and algorithms:
- Object orientation
- Abstract datatypes
- Recursion
- Searching
- Sorting
- Hashing
- Compression

Data Structures:
- Arrays
- Queues
- Stacks
- Pointers and dynamic allocation
- Lists
- Trees
- Graph (connectivity, weighted, directed)
- Network Flow

Efficiency:
- Complexity and O-notation
- Use of time and space

Teaching Methods:
Lectures
Exercises
Tutoring

Form(s) of Assessment:
Written exam, 5 hours

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Graded by course instructor and examiner.

Re-sit examination:
Re-sit examination in August.

Tillatte hjelpemidler:
A: Alle trykte og håndskrevne hjelpemidler tillatt. Alle kalkulatorer tillatt.

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Universitetslektor Frode Haug

Teaching Materials:
Textbook announced at the beginning of the semester.
Faglærer. Kompendium. Gjøvik.
Faglærer. Annet utdelt litteratur/artikler/notater. Gjøvik.

Publish:
Yes

IMT2531 Graphics Programming - Study plans 2016-2017

Course code:
IMT2531

Course name:
Graphics Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Other

Language of instruction:
English

On the basis of:

- IMT1031 Fundamental Programming
- REA1121 Mathematics for Programming

Expected learning outcomes:
On completion of this course the students will be able to:

Knowledge

- List and describe the components of the graphics pipeline.
- Understand the mathematical foundations of computer graphics
- Describe the process of Ray Tracing to create a image.
- Explain the fundamental aspects of animation with respect to programming graphics.

Skills

- Create 2D procedural animation.
- Manipulate 3D models with loading, saving and onscreen display.
- Create and manipulate lighting in a 3D scene
- Use OpenGL for rendering 3D environments
- Ask better questions about what is required for a graphical effect

General Competence

- Present the solution to a defined problem orally, and answer question about the solution
- Read and integrate academic material from various online sources
- Improved software development ability
- Reinforce version control and static code analysis
- Improvement in asking quality questions

Topic(s):

- 3D Mathematics
- 2D Graphics
- Animation
- 3D Graphics Pipeline
- Data representation for graphics
- Lighting and Textures
- Vertex buffer objects and Pixel buffer objects
- Management of graphical assets
- Advanced surface descriptions including normal mapping
- Ray Tracing
- WebGL
- Shadows and reflections

Teaching Methods:
Lectures
Exercises

Teaching Methods (additional text):
Student will work in C++ using OpenGL and SFML/SDL, in a problem based learning approach.

Form(s) of Assessment:
Home exam, 72 hours
Evaluation of Project(s)

Form(s) of Assessment (additional text):
The course will be evaluated with 40% on two internal projects and 60% on a 3-day take home exam which will conclude with an oral presentation and questions and answers.

Both parts must be passed.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Internal examiner, together with external examiner once every five years, next time in 2021.

Re-sit examination:
Re-sit examination possible, in agreement with the course responsible.

Tillatte hjelpemidler:

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Associate Professor Simon McCallum

Teaching Materials:
Online Resources plus required text:

- **Anton's OpenGL 4 Tutorials** [Kindle Edition]

with recommended texts:

- **OpenGL Programming Guide: The Official Guide to Learning OpenGL** , Version 4.3 (8th Edition)
- **OpenGL superbible : comprehensive tutorial and reference** , Richard S. Wright, 5th Ed.

Additional information:
If there are fewer than 5 students in the course the form will change to suit the class size.

Publish:
Yes

IMT2571 Data Modelling and Database Systems - Study plans 2016-2017

Course code:
IMT2571

Course name:
Data Modelling and Database Systems

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
Norsk, alternativt engelsk

On the basis of:
IMT1031 Fundamental Programming and IMT1082 Object-Oriented Programming, or IMT1241 Fundamental Programming in Java

Expected learning outcomes:
After successful completion, the student is capable of describing:
- database tasks and purpose in applications and computer systems
- database management systems; their role and tasks
- relational databases; their foundation and characteristics
- other types of database management systems (NOSQL)
- structuring and management of semi-structured data (XML)

After successful completion, the student posesses the skills to:
- assess the use of relational database, NOSQL database, and XML in actual applications and computer systems
- develop and evaluate functional and effective conceptual data models - and corresponding logcial, relational models - for real applications
- construct relational database solutions - and select appropriate physical structure - based on the conceptual and logical models designed for the system
- make use of SQL for inserting, querying, and modifying database data
- develop applications that retrieves and stores data in databases
- utilize XML technologies for storing and processing semi-structured data

The student has acquired general competency in:
- developing abstract models and solutions for practical problems
- assessing alternatives for storing and managing digital data
- using computer tools for developing software systems

Topic(s):
- Databases and database management systems
- Introduction to conceptual data modelling
- The relational model, relational algebra, and SQL
- Database design
- Database normalisation
- Query processing
- Data integrity
- Transaction management
- File organisations and indexes
- Security
- NOSQL databases
- XML data, XML DOM, XPath og XML Schema
- Data transformation

Teaching Methods:
Lectures
Laboratory work
Mandatory assignments

Form(s) of Assessment:
Written exam, 5 hours

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Internal examiner. An external examiner will be involved in the examination at least every fifth year - next time no later than 2020.

Re-sit examination:
Ordinary re-sit offered in August.

Tillatte hjelpemidler:

Coursework Requirements:
5 out of 6 assignments must be passed.

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Professor Rune Hjelsvold

Teaching Materials:
- T. Connolly & C. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management. 5th Edition, Addison Wesley 2010
ISBN-10: 0-321-52306-7
- Web resources (titles to be announced at the start of the course)

Additional information:
Overlaps 90% with IMT2261

Publish:
Yes

IMT2581 Rapid Prototyping and Innovation - Study plans 2016-2017

Course code:
IMT2581

Course name:
Rapid Prototyping and Innovation

Course level:
Bachelor (syklus 1)

ECTS Credits:
5

Duration:
Autumn and spring

Duration (additional text):
Whole academic year. This course runs from August to June.

Language of instruction:
English

On the basis of:
- IMT1031 Introductory Programming
- IMT1362 Experience Design

Expected learning outcomes:
After successfully completing the program, students will possess the following:

Knowledge
- Understand the process of rapid prototyping given severe time limitations
- Understand at least two different rapid prototyping tools
- Recognise situations or behaviours that limit innovation

Skills
- Are able to analyse a problem quickly and generate innovative solutions, which can be prototyped.
- Are able to plan the development of a prototype within limited time
- Are able to select features which can be implemented quickly
- Demonstrable skill in developing a prototype to demonstrate an innovative idea
- Are able to present the results of a prototyping session orally
- Are able to conduct an after action review of the innovation and prototyping process to identify strengths and weaknesses of their innovation processes.

General competence
- Improvement in the ability to work in a diverse team
- Improved confidence in the ability to create innovative content
- Understanding of the value of other disciplines in the development of innovative solutions

Topic(s):
The topics include, but are not limited to:

- Software innovation
  - Supporting software innovation
  - Identifying high value innovation
  - Return on investment
- Rapid prototyping tools
- Principles of rapid prototyping
  - Communication
  - Organisation
  - Preparation
- GameJams / idea24plus - idea generation and prototyping
- Prototyping in context
  - Value chains
  - Getting innovation to market
- Review and analysis of intense development cycles
  - Post mortems
  - After action review
  - Accurate performance evaluation and feedback

Teaching Methods:
Lectures
Group works
Project work
Tutoring

Teaching Methods (additional text):
This course is focused on rapid prototyping in a very time limited situation. The course requires the development of an idea, and a prototype of that idea within a period of 48 to 72 hours. The course requires the student to participate in three rapid prototyping sessions during the course. These can be selected from: Idea24, Game Jams, Global Game Jam, Games for Health Jam, The Gathering Creative Ticket, NFI Game play, or other events which support the process of rapid prototyping with extreme time pressure.

These intense development cycles are supported by lectures introducing the principles and tools of rapid prototyping, and by review sessions focused on analyzing the development cycle and learning as much as possible from each session.

Form(s) of Assessment:
Oral exam, group
Evaluation of Project(s)

Form(s) of Assessment (additional text):
Successful completion of three rapid prototyping events. Two of these will be done in groups and one individual. Each student will present their work at the end of each session either as part of the group or individually. Students will pass or fail on each game development session based on criteria which include:

- Understanding requirements
- Approach to breaking up the development tasks
- Ability to develop a prototype of an idea
- Depth of reflection during review sessions
- Identification of important events and learning opportunities.

Grading Scale:
Pass/Failure

External/internal examiner:
Each presentation will be to a team of reviewers. The exact distribution of the review team will vary, but ideally it will have:

- The supervisor for the course
- An internal academic reviewer
- A subject matter expert for the case of serious games or modeling
- A current industry professional

Re-sit examination:
Students who do not pass 3 rapid prototyping sessions within a year can participate in the same events in the following year. Once 3 have been passed in total the student will pass the course.

Tillatte hjelpemidler:

Academic responsibility:
Faculty of Computer Science and Media Technology

Emneansvarlig kobling:
Simon J R McCallum

Course responsibility:
Associate professor Simon McCallum

Teaching Materials:
Reading lists and content will be provided before each development session

Recommended reading includes:
- The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Eric Ries, 2011

Additional information:
Presentation material and notes can be brought into the oral presentation.

Publish:
Yes

IMT2243 Software Engineering - Study plans 2016-2017

Course code:
IMT2243

Course name:
Software Engineering

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Vår

Language of instruction:
Norwegian

Expected learning outcomes:
The candidate have knowledge of plan-driven and agile methodologies in software development and understand basic administrative and technological aspects of the specification, development, testing and maintenance. They know the basic principles in software architecture and design and the value of user participation.
The candidate can apply object-oriented methods and techniques of requirements specification and are able to establish project procedures using agile development methodology. They can work from project idea to a recommended sketch for a software solution in small projects and know the benefit of tools in different parts of the software development process.
The candidate gain awareness of the software's role in business and community and the role of management, teamwork and documentation in software projects.

Topic(s):
The role of software applications in companies.
Plan-driven and agile software development methodologies

Project management and risk analysis
Methods and techniques in requirement specification and analysis (UML)
Principles in Architecture, Design and Testing
Tools and Configuration management
User participation

Teaching Methods:
Lectures
Project work
Tutoring

Form(s) of Assessment:
Written exam, 3 hours
Evaluation of Project(s)

Form(s) of Assessment (additional text):
Written Exam, 3 hours (counts 40%)
Evaluation of Project(s) (counts 60%)
Each part must be individually approved of.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

Re-sit examination:
Re-sit for the written exam in August.

Tillatte hjelpemidler:
D: Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Coursework Requirements:
None

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Tom Røise

Teaching Materials:
Software Engineering, Ian Sommerville, 10th ed

Additional materials will be available at semesterstart.

Publish:
Yes

IMT2282 Operating Systems - Study plans 2016-2017

Course code:
IMT2282

Course name:
Operating Systems

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Vår

Language of instruction:
Norwegian

On the basis of:
Algorithmic methods

Expected learning outcomes:
The students will be aquainted with basic principles and methods in modern operating systems and how they are organized. This will show how a computer can optimize use of the resources. This knowledge shall help the student in evaluation, use and maintainance of operating systems.

Topic(s):
System calls, processes and threads, how they can be synchronized and how they can communicate.
CPU - scheduling algorithms.
Memory management:Virtual memory, swapping, paging and segmentation.
File systems: Implementation, backup, consistens and performance.
IO systems: Polling, interrupt and DMA. interrupt handlers, drivers, device independant layer, disk systems and timers.
Deadlocks: Detection and recovery, prevention and avoidance.
Virtualization.
Security: Access Control and Malware
Programming in C, Bash, PowerShell

Teaching Methods:
Lectures
Group works
Laboratory work
Exercises

Teaching Methods (additional text):
Lectures

3 projects
Case-study
Homework

Form(s) of Assessment:
Written exam, 5 hours

Form(s) of Assessment (additional text):
Written Exam, 5 hours

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
See Norwegian version.

Re-sit examination:
Re-sit in August.

Tillatte hjelpemidler:
D: Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Coursework Requirements:
3 mandatory assignments and 3 multiple choice tests

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Førsteamanuensis Erik Hjelmås

Teaching Materials:
Tanenbaum, A. S and Bos, H. Modern Operating Systems, 4th edition, Pearson Education, 2015.

Publish:
Yes

IMT3591 Artificial Intelligence - Study plans 2016-2017

Course code:
IMT3591

Course name:
Artificial Intelligence

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Vår

Language of instruction:
English

Prerequisite(s):
IMT1031 Fundamental Programming

On the basis of:
IMT2021 Algorithmic Methods

Expected learning outcomes:
On successful completion of the module, students will be able to

- Understand and evaluate various core techniques and algorithms of AI, namely agent technology, informed and uninformed tree and graph search algorithms, various learning techniques including artificial neural networks, decision tree learning and evolutionary algorithms, logic and planning techniques and algorithms, knowledge representation, the meaning of concepts such as intelligence, reasoning, and making inferences.
- Identify different uses and applications of AI techniques and algorithms, from neuroscience, understanding brain to game development, to web technologies and secure system designs.
- Implement several of the algorithms on the mobile robots. The students will also enhance their programming skills in a preferred language of their own and in Java by learning to program a mobile robot.
- Improve programming skills through the programming of mobile robots. Programming mobile robots help with connecting the theory learnt in class with the practical use of it.
- Evaluate the run-time and memory complexity of several AI algorithms, and practice with creating better algorithms.

Topic(s):

- Path finding
- FSM
- Scripts
- Symbolic AI Techniques
- Logic
- Multi agent systems
- State based search
- Goal directed search
- Genetic Algorithms / Programming
- Neural networks
- Reinforcement learning

Teaching Methods:
Lectures
Exercises

Teaching Methods (additional text):
This course will focus on practical implementation of AI concepts. Lectures will introduce a topic area, and students are expected to implement and report on the key concept.

Form(s) of Assessment:
Exercises
Written exam, 4 hours

Form(s) of Assessment (additional text):
Written exam, 4 hours (60%)

4 compulsory assignments (40%). Each of these assignments must be passed individually to be able to take the written exam.

Both parts must be passed.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Internal examiner

Re-sit examination:
Re-sit examination in August for the written exam.

The assignments must be taken the next time the course is running.

Tillatte hjelpemidler:
A: Alle trykte og håndskrevne hjelpemidler tillatt. Alle kalkulatorer tillatt.

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Associate Professor Sule Yildirim

Teaching Materials:
Artificial Intelligence: A Modern Approach, 3rd Edition by Stuart Russell and Peter Norvig, 2010

Additional information:
In case there will be less than 5 students apply for the course the form may change to suit the class size.

Publish:
Yes

IMT3601 Game Programming - 2015-2016

Course code:
IMT3601

Course name:
Game Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
English

Prerequisite(s):

- IMT2531 Graphics Programming or IMT3591 Artificial Intelligence

On the basis of:

- IMT1361 Game Design

Expected learning outcomes:
On completion of this course the students will be able to:

Knowledge

- Understand fundamentals of game engines
- Understand the social and ethical issues in game development
- Understand continuous integration and unit testing in realtion to game development.

Skills

- Design and implement a computer game as part of a group
- Better estimate the amount of effort required to implement various features in a computer game
- Use project management tools to control the development of software
- Conduct code reviews referring to static code analysis and profilling tools
- Justify development decisions based on evidence from sources including textbooks and the Internet
- Gui prototyping of interactions.

General Competence

- Communicate about the development process and present the results both in written and oral form

This is a group project and so a significant part of the learning outcomes relate to the working in a group and being able to scope the time taken to implement a game design.

Topic(s):
The lectures will be of the "Just in time" variety, where the topics will focus what the students need to know to continue the development of the project. The topics will include:

- Design patterns
- Graphics
- Physics in games
- Character development
- Animation
- Game specific AI
- Implementing game mechanics
- Game production process
- Project management in teams
- Memory management
- Algorithm efficiency
- C++ techniques

Teaching Methods:
Lectures
Exercises

Teaching Methods (additional text):
Student will work in groups to develop a game within various design constraints. In class exercises and lectures will be linked to the current stage of the development process.

Form(s) of Assessment:
Evaluation of Project(s)

Form(s) of Assessment (additional text):
One large project, which will include regular presentations of progress, and a presentation of the game in the week before final delivery.

In cases where groups breakdown and are unable to work together students will be given an oral exam to assess their ability and learning.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
External examiner once every three years, with an internal examiner in other years. First year for external 2014.

Re-sit examination:
None

Tillatte hjelpemidler:

Examination support:
No examination

Academic responsibility:
Faculty of Computer Science and Media Technology

Emneansvarlig kobling:
Simon J R McCallum

Course responsibility:
Associate Professor Simon McCallum

Teaching Materials:
Game Coding Complete, Fourth Edition (2012), Mike McShaffy

Additional information:
In case there are fewer than 5 students which apply for the course the form of presentation and assessment may change to suit the class size.

Publish:
Yes

Home page:
Game Programming

IMT3662 Mobile Development Theory - 2015-2016

Course code:
IMT3662

Course name:
Mobile Development Theory

Course level:
Bachelor (syklus 1)

ECTS Credits:
5

Duration:
Autumn

Duration (additional text):
First half of semester

Language of instruction:
English

Prerequisite(s):

- IMT1031 Introduction to Programming or
- IMT1241 Basic Java Programming (from autumn 2013 replaced by IMT1441 Programming for web I).

On the basis of:

- IMT1082 Object Oriented Programming or equivalent
- IMT2291 Web Technologies or equivalent

Expected learning outcomes:
On completion of this course the students will have the following skills, knowledge and general competences:

Knowledge

- Recognize and dicsuss the strengths and weaknesses of mobile devices.
- Know the performance limitations of mobile technologies and networks.
- Discuss and review the special user interface requirements of mobile devices.
- Have a in-deepth understanding of the development process for a specific mobile platform.
- Identify the ethical and social impact of mobile devices on society.

Skills

- Be able to identify and characterize the strengths and weaknesses of mobile devices.
- Be able to classify and describe the performance limitations of mobile technologies and networks.
- Be able to use an integrated development environment (IDE) to implement applications for mobile devices.
- Be able to access and use the variety of input methods found in mobile devices.
- Develop an appreciation of the potential for innovation in mobile services, and the impact this could have on society.
- Be able to cross compile code for at least two different mobile Operating Systems: Android and iOS

General competence

- Improved understanding of the software development process
- Exposure to and understanding of group projects and group dynamics in the software development team
- Improved programming skills

Topic(s):
The field of mobile system development changes rapidly. Thus the topics covered need to be flexible. In this course these include, but are not limited to:

- Strengths and weaknesses of mobile technologies
- Limitations of mobile devices available on the market
- Programming Design Patterns for mobile systems
- Layout and UI designs and practices for mobile screens (phones, tabets, wearables)
- Using alternative input interfaces
- Sensor integration - GPS, accelerometer/gyro, tilt, magnetic field, compass and camera
- Raw data filtering and signal processing
- OpenGL ES - 2D/3D graphics on mobile devices
- Managing multiple product SKU's
- OS specific development issues: Apple's iOS and Google's Android
- Mobile networking and technology stack

Teaching Methods:
Lectures
Net Support Learning
Project work

Teaching Methods (additional text):
Student will receive an overview of the mobile development process on a range of devices, but will select one for in depth study. Some mobile devices will be available for testing, but it would be beneficial for the student to have access to a personal mobile device (for example iOS which requires an iOS-enabled device, or and Android 4.2+)

Form(s) of Assessment:
Other

Form(s) of Assessment (additional text):

- Portfolio of internal assignments (60%).
- Final, written exam 3 hours (40%)

  There are 3 internal assignments that will be contributing to the final 60% of the course mark. The first assignment is a simple application demonstrating the use of activities, and simple GUI elements. The second assignment is an individual assignment focused on developing a simple app which accesses both sensor data and network connectivity and use of some Internet resources. The final third assignment is a small, group project implementing a new or extending an existing application, such as a simple game, visualization tool, media production app, or productivity tool. All internal assignments count for 60% of the total mark, and each of them needs to be passed in order to be able to attend final written exam. The group project will be on a phone/mobile device of the students choice.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Internal examiner, together with external examiner once every three years on the written exam, next time in 2016.

Re-sit examination:
Re-sit August 2016 for the written exam.

Tillatte hjelpemidler:

Examination support:
None

Coursework Requirements:
Passing the internal portfolio is required to sit the exam.

Academic responsibility:
Faculty of Computer Science and Media Technology

Emneansvarlig kobling:
Mariusz Nowostawski

Course responsibility:
Associate Professor Mariusz Nowostawski

Teaching Materials:
Web resources from Apple, Google, Microsoft and other mobile device operators as well as other online tutorial sites.

Replacement course for:
IMT3661

Publish:
Yes

Elective course, 10 ECTS - 2015-2016

Course name:
Elective course, 10 ECTS

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn and spring

Language of instruction:
Norwegian

Expected learning outcomes:
.

Topic(s):
.

Teaching Methods:
Group works

Form(s) of Assessment:
Exercises

Grading Scale:
Pass/Failure

Tillatte hjelpemidler:

Academic responsibility:
Faculty of Technology, Economy and Management

Course responsibility:
.

Publish:
Yes

IMT3912 Bachelor's thesis - 2015-2016

Course code:
IMT3912

Course name:
Bachelor's thesis

Course level:
Bachelor (syklus 1)

ECTS Credits:
20

Duration:
Autumn and spring

Language of instruction:
Norwegian

Prerequisite(s):

- The students must pass 100 ECTS credits before they can start working on the bachelor thesis.
- From autumn 2013: Idélab 24 or similar

Expected learning outcomes:
The bachelor thesis concludes the study and will combine important parts of the scientific content of the study program. After successful completion the students have acquired:

Knowledge:

- new knowledge in a part of the subject area chosen by the students
- an understanding of how to work in systematic ways, having reflective capabilities, and being able to conduct systematic/scientific assessments
- knowledge of the process of planning and conducting an independent piece of work, specifying problem to solve and analysing the identified problems based on theoretical as well as empirical data, and solve the problem in a methodologically acceptable way

Skills:

- skills in composing a problem statement of general interest within the subject area, under supervision
- skills in searching and identifying relevant scientific literature, under supervision
- skills in studying a delimited problem and developing alternative ways to solve the problem
- skills in documenting and presenting project results in a systematic/scientific way

General Competence:

- appreciation of scientific ethics; being capable of identifying ethical concerns of relevance for the chosen problem
- a consciousness of the impact the project may have on individuals, companies, and society

Topic(s):
The students choose a preapproved problem within the subject area.

Teaching Methods:
Project work
Tutoring

Form(s) of Assessment:
Evaluation of Project(s)

Form(s) of Assessment (additional text):
For students attendiing Bachelor of Science in Engineering - Computer Science:

- The project report is given a temporary grade.
- Individual oral examination/presentation may adjust the grade up or down to the final grade, according to performance.
- Students must obtain a passing grade on the report to be able to present themselves for the oral examination/presentation.
- Students must pass both parts to pass the course
- For off campus students the oral exam will be arranged through web conference.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
External and internal examiner

Re-sit examination:
After failure, a student may submit a new or a revised thesis once. If the student chooses to submit a revised version of the thesis, this must be submitted in the following semester.

Tillatte hjelpemidler:

Coursework Requirements:

- Problem description
- Project plan
- Written report signed by all project members
- Individual reflection notes
- Project presentation on Internet
- Project poster
- Oral presentation at the end of the project

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Hilde Bakke

Publish:
Yes

IMT3102 Object-Oriented Software Development - 2015-2016

Course code:
IMT3102

Course name:
Object-Oriented Software Development

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
Norwegian

Expected learning outcomes:
Students who have passed this course should have an in-depth understanding of incremental and iterative software methodologies. They will be able to specify, analyse and design software based on object-oriented modelling supported by UML (Unified Modelling Language). Furthermore they are introduced to the use of Patterns in software design.

Topic(s):
Incremental and iterative software methodologies
Object-oriented analysis
Object-oriented design
Unified Modelling Language
Architecture and Design Patterns
Use of development-tools

Teaching Methods:
Lectures
Group works
Project work
Reflection
Tutoring

Form(s) of Assessment:
Portfolio Assessment

Form(s) of Assessment (additional text):
In this course three group deliveries and three individual deliveries are required. The final assesment is based on four of these deliveries.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

Tillatte hjelpemidler:

Coursework Requirements:
None

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Høgskolelektor Tom Røise

Publish:
Yes

IMT3281 Software Development - 2015-2016

Course code:
IMT3281

Course name:
Software Development

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
Norwegian

Prerequisite(s):

- IMT1031
- IMT1082

On the basis of:

- IMT2021

Expected learning outcomes:
Knowlegde

- The candidate should be able to describe the principles behind and the structure of general distributed systems.
- The candidate should be able to use ready-made modules as well as design and use class libraries.
- The candidate should be able to organize the functionality in the appropriate class and package structures.

Skills

- The candidate should be able to use existing libraries to produce complex multi-threaded program systems.
- The candidate should master the development of GUI applications with multiple windows
- The candidate should master the use of development tools and version control systems.

General competence

- The candidate can plan and carry out software development projects.
- The candidate may use relevant interaction systems that provide the opportunity to work together on projects even if the participants are located in geographically different locations.
- The candidate will through the work in this course acquire good and practical skills in programming.

Topic(s):

- Class libraries, development and use
- Multithreaded systems
- Window based applications
- Distributed programming
- Usage of databases and XML
- Source code documentation
- Usage of development tools and version control systemts

Teaching Methods:
Lectures
Laboratory work
Project work

Form(s) of Assessment:
Other

Form(s) of Assessment (additional text):
Written Exam, 4 hours (counts 45%, evaluated by lecturer)
Evaluation of Project(s) (counts 55%, evaluated by lecturer)
Two projects: One large project counts 45 %. One smaller project counts 10 %.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
The person responsible for the course will grade both projects and the written exams
An external examiner will be used in addition for the written exams every fourth year, next will be in 2014

Re-sit examination:
Re-sit August 2016 for the Written exam

Tillatte hjelpemidler:

Examination support:
All written materials

Coursework Requirements:
None

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Høgskolelektor Øivind Kolloen

Teaching Materials:
Java How to Program, Eighth Edition, Deitel/Deitel, Prentice Hall, 2010

Publish:
Yes

IMT3861 Stormaskiner - 2015-2016

Course code:
IMT3861

Course name:
Stormaskiner

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Autumn

Language of instruction:
Norwegian

On the basis of:
IMT2282 Operativsystemer
IMT2431 Datakommunikasjon og nettverkssikkerhet

Expected learning outcomes:
Knowledge

- Explain the concepts of the mainframe architecture, basic functionality of z/OS, mainframe's area of use in industry
- Explain how file processing and program execution conceptually differs between mainframes and PC architecture

Skills

- Master the JCL language
- Write and execute batch jobs on z/OS
- Perform basic application development on z/OS
- Implement basic databases on DB/2
- Implement web applications on WebSphere on z/OS

General competence

- Justify how high availability systems like mainframes perform important IT services in society

Topic(s):

- Mainframe architecture
- High availability
- z/OS and z/OS tools
- Data set and files
- JCL (Job Control Language)
- Batch programming
- Application development on z/OS
- Transaction management and database systems on z/OS (DB2)
- Application servers on z/OS
- System programming, security and networks on z/OS

Teaching Methods:
Lectures
Laboratory work
Mandatory assignments
Exercises

Form(s) of Assessment:
Written exam, 3 hours

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Vurderes av intern sensor.

Re-sit examination:
Re-sit August 2016

Tillatte hjelpemidler:

Examination support:
Ingen

Coursework Requirements:
En obligatorisk oppgave som er en skriftlig innlevering som består av en samling av praktiske laboratoriearbeider og teoretiske øvinger må være godkjent for adgang til eksamen.

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Førsteamanuensis Erik Hjelmås

Teaching Materials:
IBM Redbooks. Introduction to the New Mainframe: z/OS Basics. Vervante, 2011.

Tilleggsartikler vil bli utdelt.

Additional information:
Studentene må ha med egen laptop på alle forelesninger og øvinger.

Publish:
Yes

IMT3672 Mobile Development Project - 2015-2016

Course code:
IMT3672

Course name:
Mobile Development Project

Course level:
Bachelor (syklus 1)

ECTS Credits:
5

Duration:
Autumn

Duration (additional text):
Second half of the semester

Language of instruction:
English

Prerequisite(s):

- IMT1031 Introduction to Programming or
- IMT1241 Basic Java Programming

On the basis of:

- IMT1291 Web Design
- IMT3662 Mobile Development Theory (or IMT3661) or
- IMT2551 Mobile System Fundamentals

Expected learning outcomes:
The goal of this course is to provide students with experience in developing complete mobile applications.

- the project is conducted in groups
- the focus is on in-depth understanding of mobile development process, from an idea to a final deployed/distributed product
- we engage with advanced mobile programming and UI techniques
- we stress the social nature of mobile applications
- each application will use a range of sensors and advanced programming techniques
- the project aim is to form the basis and expose students to an entrepreneurial activity

In addition to the above, this elements are taken into consideration

- communication and team dynamics
- software engineering processes
- innovation: startups and indie development teams

At the end of the project the students will have:

Knowledge

- ability to recall main mobile application components, their function, and purpose
- in-depth understanding of programming techniques for mobile applications
- ability to recognize and plan the mobile applications structure
- in-deepth knowledge of mobile development process

Skills

- improved ability to work in a group and present the results of a project
- ability to develop a mobile solution to a defined problem
- ability to implement a mobile application and utilize the screen, input mechanisms, database, social aspects and Internet connectivity characteristic to mobile platforms
- improved experience and ability to conduct user testing
- improved experience and ability to participate in iterative development

General competence

- understanding and improved skills in development process from an idea to a fully finished, deployed product
- improved entrepeneurial skills and understanding of the process

Topic(s):
The students will choose the topic for the project in consultation with the supervisor. The choice of supervisor will depend on the topic and the application target platform. The devices that projects target are:

- Android devices (phones and tablets)
- Apple iOS mobile devices (iPad, iPhone and iPod)
- other (wearable or ubiquitous technologies) subject to project themes and supervision

Having chosen a platform the students may propose a project or select from a list of research projects provided by the supervisors for each device.

The project is expected to be innovative and completed/polished by the end of the course. The use the core features of the mobile technologies as well as social elements are encouraged. Unless agreed with the supervisor, the application is expected to be a native application on a target platform.

Teaching Methods:
Project work

Teaching Methods (additional text):
Student will form groups of 3-4 with an encouragement to create a multidisciplinary team: a mix of students from programming and media focused degrees. Student will receive course supervision from the course coordinator as well as additional supervision from faculty with experience with the chosen platform. The groups will be expected to track their progress through version control and issues tracking system and provide regular progress reports.

It is **strongly** encouraged for students to own their own Android or iOS device when taking this course.

Form(s) of Assessment:
Oral exam, group
Evaluation of Project(s)

Form(s) of Assessment (additional text):
75% project and 25% oral presentation.

The students will present their project at the end of the course. Both parts of the course must be passed independently.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
A mix of internal and external examiners based on the projects chosen.

Re-sit examination:
There is no re-sit examination.

Tillatte hjelpemidler:

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Associate Professor Mariusz Nowostawski

Teaching Materials:
There is no textbook for this course. However, web based resources from Apple and Google will provide the basis for the technical skills required, with additional textbooks and online resources available for each platform on per need basis.

Replacement course for:
IMT3671

Publish:
Yes

IMT3801 Multi-threaded Programming - 2015-2016

Course code:
IMT3801

Course name:
Multi-threaded Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
5

Duration:
Autumn

Language of instruction:
English

Prerequisite(s):

- IMT2021 Algorithmic Methods

Expected learning outcomes:
On completion of this course the students will be able to:Understand the classic examples of concurrency problems

- Understand the benefits and difficulties of developing multi-threaded applications
- Demonstrate knowledge of thread-safe implementations of design patterns commonly used in game development
- Evaluate the performance of multi-threaded and optimised programs
- Communicate the design decisions, data flow and synchronisation of parallel algorithms in both written and oral form

Topic(s):

- Multi-threaded programming
- Profiling and performance evaluation
- Algorithmic complexity
- Design Patterns
- Function Pointers and Callbacks
- Game programming techniques
- Distributed programming environments

Teaching Methods:
Lectures
Laboratory work

Teaching Methods (additional text):
Student will work primarily in C++ using various parallel programming libraries, in a problem based learning approach.

Form(s) of Assessment:
Oral exam, individually
Evaluation of Project(s)

Form(s) of Assessment (additional text):

- Portfolio of internally completed work 60%
- Oral exam 40%
- Both parts must be passed

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Projects evaluated by lecturer.
Oral exam will have two internal examiner and we will use external examiner every fifth year, next time 2016.

Re-sit examination:
Oral exam may be repeated

Tillatte hjelpemidler:

Examination support:
None

Coursework Requirements:
None

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Jayson Mackie

Teaching Materials:
No required textbooks. Books, monographs, and research articles will be recommended during the course.

Additional information:
In case there are fewer than 5 students which apply for the course the form of presentation and assessment may change to suit the class size.

Publish:
Yes

IMT2291 Web Technology - 2015-2016

Course code:
IMT2291

Course name:
Web Technology

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Vår

Language of instruction:
Norwegian

Prerequisite(s):
IMT1031 **or** IMT1241

On the basis of:
IMT1082 or IMT1241
IMT1041

Expected learning outcomes:
Knowledge

- The candidate should be able to evaluate different solutions and make reasoned choices for the development of advanced web based applications.
- The candidate should be able to plan and organize the development of web-based applications.
- The candidate should be able to divide a web application in different layers.

Skills

- The candidate should be able to run a web development project and implement a final solution based on a customer's needs.
- The candidate should be able to combine different technologies and principles to create new solutions.
- The candidate should be able to further develop existing products to meet new demands

General competency

- The candidate has knowlegde about central interaction systems that allow work in groups both locally and at distance.

Topic(s):

- The HTTP-protocol
- Serverside programming i PHP
- Variables through HTTP, cookies and session management
- Database usage (MySQL)
- HTML/Javascript/CSS
- DOM
- Ajax
- Dynamic web interfaces

Teaching Methods:
Lectures
Laboratory work
Project work

Form(s) of Assessment:
Written exam, 3 hours
Evaluation of Project(s)

Form(s) of Assessment (additional text):
Written Exam, 3 hours (counts 60%, evaluated by lecturer)
Evaluation of Project(s) (counts 40%, evaluated by lecturer)
Each part must be individually approved of.
There will be 2 projects in the course, each accounts for 20% of the final grade.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
The person responsible for the course will grade both projects and the written exams
An external examiner will be used in addition for the written exams every fourth year, next will be in 2014

Re-sit examination:
Re-sit August 2016 for the Written exam.

Tillatte hjelpemidler:

Examination support:
B: All printed and hand-written support material is allowed. A specific basic calculator is allowed

Coursework Requirements:
None

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Høgskolelektor Øivind Kolloen

Teaching Materials:
Ajax in action, Dave Crane/Eric Pascarello, Manning, 2006
PHP5 and MySQL Bible, Tim Converse/Joyce Park, Wiley Publishing, Inc., 2004

Publish:
Yes

IMT3511 Discrete Mathematics - 2015-2016

Course code:
IMT3511

Course name:
Discrete Mathematics

Course level:
Bachelor (syklus 1)

ECTS Credits:
10

Duration:
Spring and autumn

Duration (additional text):
In principle this course will be given in the spring semester, but in case there is enough interest, then it can also be given in the fall semester.

Language of instruction:
English

Expected learning outcomes:
Knowledge:

- The candidate possesses knowledge of important topics within abstract algebra.
- The candidate possesses knowlegde of important topics within combinatorics.
- The candidate possesses knowlegde of fundamental topics within graph theory.

Skills:

- The candidate knows relevant methods and terminology in discrete mathematics.
- The candidate is capable of applying his/her knowledge in different courses.

General competence:

- The candidate is capable of understanding and analyzing problems related to abstract algebra, combinatorics and graph theory.

Objectives:
After the course, the students should acquire:

- Understanding of the most important topics of abstract algebra
- Understanding of the most important topics of combinatorics, including fundamentals of graph theory.

Topic(s):
General concepts:
* Logic, proofs, sets, algorithms, induction and recursion, combinatorics, discrete probabilities

Graphs:
* Connectivity, shortest path, (minimal) spanning trees

Modeling computation:
* Finite-state machines, Turing machines

Abstract algebra:
* Groups, rings, fields

Teaching Methods:
Lectures
Exercises
Tutoring

Teaching Methods (additional text):
The course is given as a self reading course, where there is time for the students during lectures to raise questions on the theory and/or the exercises.

The course will be made accessible for both campus and remote students. Every student is free to choose the pedagogic arrangement form that is best fitted for her/his own requirement. The lectures in the course will be given on campus and are open for both categories of students. All the lectures will also be available on Internet through GUC's learning management system (ClassFronter).

Form(s) of Assessment:
Oral exam, individually

Form(s) of Assessment (additional text):
Candidates will get an oral exam (max 45 minutes) with written preparation (max 60 minutes). Candidates will be given a number of assignments within the topics of the course and 60 minutes to prepare written answers. After this the candidates will be questioned about their answer in the oral part.

If the number of students is too high, then the oral exam is replaced by a 3 hour written exam. The students will be notified about this one month prior to the exam at the latest.

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Evaluated by either internal and external examiner or 2 internal examiners.

Re-sit examination:
Re-sit August 2016

Tillatte hjelpemidler:

Examination support:
D: No printed or hand-written support material is allowed. A specific basic calculator is allowed.

Coursework Requirements:
None.

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Professor Patrick Bours

Teaching Materials:
- Kenneth H. Rosen:
Discrete Mathematics and its Applications, 7th ed.
McGraw-Hill International Edition (2012), ISBN 978-0-07-338309-5.
- William J. Gilbert and W. Keith Nicholson
Modern Algebra with Applications, 2nd ed.
Wiley (2004), ISBN 0-471-41451-4

Additional information:
In case there will be less than 5 students that will apply for the course, it will be at the discretion of Studieprogramansvarlig whether the course will be offered or not an if yes, in which form.

Publish:
Yes

IMT3612 GPU Programming - 2015-2016

Course code:
IMT3612

Course name:
GPU Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
5

Duration:
Vår

Duration (additional text):
Spring

Language of instruction:
English

Prerequisite(s):
IMT2531 Graphics Programming

Expected learning outcomes:
After completing the subject the students should improved

Knowledge

- Understanding the archetecture of a GPU
- Understand the role of visual effects in games and their connection to player experience
- Understand how to use a GPU as a general proceesing device

Skills, the student will be able to:

- Program the graphics processor (GPU), i.e. write shaders
- Use software for testing and development of shaders
- Develop a GPU shader to implement a relevant feature in a computer game.
- Impement a non graphics specific algorithm on a GPU

General competence

- Improved ability to analyse a problem and find a parallel solution
- Improved general programming ability
- Improved process around testing and assessing code.

Topic(s):
Topics will include but are not limited to:

- GLSL - API and language
- Lights, materials and textures
- Raycasting
- Use of multiple shaders

Teaching Methods:
Lectures
E-learning
Mandatory assignments

Teaching Methods (additional text):
Online Lectures
Local tutorials
Local Assignments

Form(s) of Assessment:
Home exam, 24 hours

Form(s) of Assessment (additional text):
24 hour take home exam with a 15 minute oral discussion

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
Two internal examiners, or internal and external examiners. External examiners next time in 2015.

Re-sit examination:
Resit exam within 2 months of orignal exam based on the same format.

Tillatte hjelpemidler:

Coursework Requirements:
2 assignments

Academic responsibility:
Faculty of Computer Science and Media Technology

Course responsibility:
Førsteamanuensis Simon McCallum

Teaching Materials:
GPU computing at http://www.gpucomputing.net/

Open textbook: Programming on Parallel Machines; GPU, Multicore, Clusters and More, Norm Matloff, http://heather.cs.ucdavis.edu/parprocbook

Udacity course: https://www.udacity.com/course/cs344

Notes : GLSL-tutorial from Lighthouse3D.com

Replacement course for:
IMT3611

Additional information:
The IMT3611 course was 10 points, this content has been separated into IMT3801 Multithreaded Programming and this 5 point verison.

Publish:
Yes

IMT3602 Professional Programming - 2015-2016

Course code:
IMT3602

Course name:
Professional Programming

Course level:
Bachelor (syklus 1)

ECTS Credits:
5

Duration:
Vår

Duration (additional text):
Spring

Language of instruction:
English

Prerequisite(s):
IMT2021 Algorithms

IMT2243 Software Engineering

On the basis of:
Working on a large full semester project in another course, for example the Bachelor Oppgave or Masters Thesis.

Expected learning outcomes:
The students will learn skills and knowledge related to developing a project using the principles of professional software development.

Knowledge:

- Understanding the strenghts and weaknesses of different programming languages
- Understanding the need for process control, and communication systems for software development

Skills:

- Use of version control systems in large development projects, including ticket tracking, branching, SKUs and deployment
- Ability to comment code in accordance with an agreed standard and in a professional manner
- The ability to program for clarity
- Develop and build library components for larger systems
- Integration of multiple libraries into a large project

General Competence:

- Professionalism in approach to software development
- Give and receive critisim of coding practices and decisions

Topic(s):
The topics include but are not limited to:

- Using version control in teams.
- Coding styles
- Comparitive languages
- Bug tracking and solving
- Commenting styles
- Deployment of applications
- Integrating libraries
- Developing library modules.

Teaching Methods:
Group works
Project work

Teaching Methods (additional text):
The main teaching method for this course will be group meetings with code reviews. Students will present their work and have that worked review in front of the group. This allows students to learn from each other, and helps students learn to present their code and defend their coding decisions

Form(s) of Assessment:
Portfolio Assessment

Form(s) of Assessment (additional text):
The assessment of this course is based on:

- Quality of code written
- Quality of comments and coding style
- Quality and relevance of comment comments in version control
- Quality of involvement in code reviews and refactoring of code

Grading Scale:
Alphabetical Scale, A(best) – F (fail)

External/internal examiner:
The course will use internals and intermitent external examiners. The externals will participate every 3 years starting in 2015-2016.

Re-sit examination:
No resit. The course must be retaken.

Tillatte hjelpemidler:

Examination support:
None

Academic responsibility:
Faculty of Computer Science and Media Technology

Emneansvarlig kobling:
Simon J R McCallum

Course responsibility:
Associate Professor Simon McCallum

Teaching Materials:
Web based resources, based on the language and processes chosen for the project.

Publish:
Yes