Curriculum
for
Computer Science (AP)

Business Academy Programme (AP)
In Information Technology

ErhvervsAkademi
Sjælland

Contents:

Computer Science AP Curriculum

1.0 Preparation and framework.

This curriculum for the AP Degree Programme in Computer Science has been prepared by the Zealand Institute of Business and Technology (ZIBaT).
It was prepared within the framework of Government regulations on Examinations, on Business Academy Programmes (AP) in Information Technology (AP in Computer Science) ***, and the Common Framework for Curricula, and is valid for students starting in the AP Degree Programme in Computer Science after 1 August 2009.

1.1 Aim of the programme

The aim of the programme is to qualify the student to carry out independent work in analysis, planning and implementation of solutions for new development, enhancement and integration of IT systems in private and public enterprises nationally and internationally.

1.2 Length of the programme

The official duration for this full-time programme is 2½ years of full-time study. A year of full-time study is a full-time student's work over one year. A year of full-time study equals 60 European Credit Transfer System (ECTS) points. The full programme thus equals 150 ECTS points.

1.3 Title of graduates

Graduates of the programme have the right to use the title **AP Graduate in Computer Science**.

2.0 Learning objectives of the programme

The learning objectives are the knowledge, the skills and the competences which an AP Graduate in Computer Science must acquire during the programme.

Knowledge:
The AP Graduate in Computer Science has knowledge of:

1. the practice, theory and methods normally used in software development.
2. basic business matters, with a view to using this knowledge in connection with system development
3. the technological concepts and the technology on which IT systems are based, with a view to using this knowledge in connection with programming, finding errors, and implementation.

Skills:
The AP Graduate in Computer Science is able to:

1. identify requirements for IT systems methodically and evaluate to what extent the requirements can be met within the given framework
2. take part in a system development process using modern methods, techniques and tools
3. construct software using modern and up-to-date programming techniques and tools, and ensure quality control of the product developed
4. document the work done in such a way that the documentation can be used by the given target group
5. combine knowledge of system development, programming and technology in connection with system development, programming and implementation
6. identify errors systematically in IT systems and correct them
7. evaluate practical problems as well as outline and select solutions
8. communicate practical problems and solutions to partners and users

Competences:
The AP Graduate in Computer Science is able to:

1. follow and take part in the practice of software development
2. follow developments in theory and methods in software development
3. play a competent role in project work
4. take part in specialist and interdisciplinary collaboration in connection with software development with a professional approach
5. acquire, in a structured way, new knowledge, skills and competences in relation to the IT sector, including:
   a) knowledge about domains and technology
   b) the use of new methods, techniques and tools

3.0 Profile and core areas of the programme

The programme consists of compulsory study elements totalling 115 ECTS points, elective study elements totalling 20 ECTS points, and a final examination project of 15 ECTS points.

The compulsory parts include a period of work experience totalling 15 ECTS points and the following core subject areas totalling 100 ECTS points:

- Programming 40 ECTS
- System development 25 ECTS
- Technology 20 ECTS
- Business 15 ECTS
- Specialisation 20 ECTS
- Work experience 15 ECTS
- Final examination project 15 ECTS

The core areas Programming and System Development are the primary subject areas, whereas Technology and Business are considered supportive subject areas that focus on the context in which System Development and Programming take place.

4.0 Learning objectives for the core areas of the programme

4.1 Programming (40 ECTS)

The Programming area has the following learning objectives:

Knowledge:

The aim is that the student

- Can understand both the qualitative and the quantitative properties of algorithms
- Knows about classical data structures, including their qualitative and quantitative properties
- Can understand specifications of abstract data types
- Knows about the criteria for program quality
- Knows something about describing the syntax and semantics of formal languages
- Knows about the abstraction mechanisms in modern programming languages

Skills:

The aim is that the student

- Can specify and construct algorithms
- Can use fundamental algorithm templates and programming techniques
- Can use central design patterns
- Can select suitable data structures for creating abstract data types
- Can use abstract data types when creating programs

- Can use programming language for creating algorithms, templates, patterns, abstractions and data structures
- Can use programming language for creating design models
- Can use the program libraries associated with the programming language
- Can use programming languages and their associated program libraries for creating user interfaces
- Can use a development environment associated with the programming language
- Can use means and tools to achieve quality programs
- Can create models in a database system
- Can construct programs that use a database interface
- Can use the data definition language and data manipulation language of a database system
- Can design and construct a program as a set of collaborating processes/threads
- Can use techniques for constructing programs with several simultaneous users
- Can design and construct a program based on a set of collaborating processes in a distributed architecture
- Can construct programs which use modern network technology
- Can use patterns for software architecture, including frameworks
- Can use and develop software components

Competences:

The aim is that the student is able to

- take part in development and maintenance projects as a competent programmer
- keep up-to-date on current programming languages and development tools
- acquire new programming and program design techniques.

4.2 System Development (25 ECTS)

The area of System Development has the following learning objectives:

Knowledge:

The aim is that the student

- can understand the significance of modelling in connection with system development
- can understand component-based development
- has understanding of system development methods
- can understand the characteristics of a project organisation
- has understanding of several different process models in project work
- can understand the relevance of experiment as a part of or supplement to system development methods
- can understand the significance of quality criteria for the system development process and the final configuration of the system

Skills:

The aim is that the student

- can use tools and techniques for the construction of relevant models
- can create models based on patterns
- can select and/or adapt a method for a given situation
- can use a current system development method
- can use techniques for the involvement of users in system development
- can analyse the system development domain with a view to understanding the business and technical situation and determine requirements
- can design suitable IT systems and can prepare and distinguish between various solutions in terms of requirements and conditions
- can design an overall architecture for both centralised and distributed systems
- can design user interfaces
- can design databases
- can organise and control small development projects
- can devise a project strategy for the selection of and/or modification of a process model to suit the given circumstances
- can use IT tools to support activity during a system development process
- can use experiments for the systematic discovery of user requirements
- can use experiments for the systematic investigation of technological options and limitations
- can ensure product and process quality control

Competences:

The aim is that the student

- can play a competent role in a development project
- can reflect on his/her own practice with regard to methods and process
- can adapt and combine process models and system development methods for a project to suit the particular situation
- can follow technological developments and continuously acquire new process models and system development methods.

4.3 Technology (20 ECTS)

The technology area has the following learning objectives:

Knowledge:

The aim is that the student

- knows about memory management and its significance when running programs
- knows about facilities in modern file systems
- can understand the organisation and running processes and threads
- knows about the structure of computers and operating systems
- has some knowledge about how the underlying layer supports constructions in the programming language selected

- knows about the principles for the building of fault-tolerant systems,
- can understand central security concepts, including authorisation, authentication, encryption and logging
- can understand the central threats of a technical character that an IT system can face, and understand how these threats can be met
- can understand the principles for the design and implementation of distributed systems
- can understand techniques for the integration of heterogeneous systems
- can understand the functionality of various types of standard servers, including web servers and application servers
- can understand a stratified communication model
- can understand addressing in a network
- knows about types of net and network components.
- can understand the facilities in and mode of operation of a modern database server, including handling of transactions,
- can understand how a database server handles enquiries.
- knows about technological developments, including trends.

Skills:

The aim is that the student

- can use mechanisms for synchronisation between threads
- can use standard components for secure communication
- can use common application protocols in the construction of distributed systems
- can analyse system architectures and can make choices between suggested solutions for a given problem
- can use the services offered by various types of standard servers, including web servers and application servers
- can use a programming interface for communication networks

Competences:

The aim is that the student

- can use his/her basic knowledge of technology in connection with system development and programming
- can follow developments in the field of technology and understand how these developments affect system development and programming

4.4 Business (15 ECTS)

The business area has the following learning objectives:

Knowledge:

The aim is that the student

- can understand organisational structures and the factors that are decisive for the organisation's construction
- can understand a company's management and business processes
- can understand different types of management, including project management and group behaviour in an organisation
- can understand processes of organisational change in connection with new IT applications.
- can understand the process of innovation
- can understand various e-business models
- can understand the significance of business and IT strategies for the company's IT systems
- knows about financial concepts and the principles and methods for registration and control of information
- knows about the characteristics, construction and mode of operation of ERP systems
- can understand how an ERP system connects with both the company's central processes and external events
- can understand how IT systems are implemented
- knows about the significance of organisational structure for IT security
- has knowledge about risk and vulnerability evaluation

Skills:

The aim is that the student

- can formulate a company's requirements for IT systems
- can prepare a Business Case
- can analyse and draw up descriptions of a company's business area and business procedures
- can take part in the preparation of plans for new IT solutions
- can prepare an IT strategy, including incorporation of e-business concepts on the basis of a business strategy
- can take part in the preparation of cost/benefit analyses and investment calculations
- can take part in the preparation of financial budgets and analyses
- can work out an ERP system's areas of application and make business adjustments
- can use models to describe a company's systems
- can describe the consequences of IT purchases
- can analyse a company's organisation of IT security

Competences:

The aim is that the student

- can follow developments in the field of technology and understand how these developments can affect and be used by companies
- can perform in various types of organisation
- is capable of contributing with business understanding in connection with the purchase, development and implementation of IT systems in accordance with a given situation
- can include relevant business aspects in connection with both strategic and operational decisions on the development and use of IT systems.

4.5 Topics and ECTS distribution in the programme's core areas.

Programming	*System development*	*Technology*	*Business*
40 ECTS	***25 ECTS***	***20 ECTS***	***15 ECTS***
Content consists of the following topics: - Algorithms - Templates, techniques and patterns - Data structures and abstract data types - Programming languages - Program quality - Language theory - Database programming - Concurrency - Distributed programming - Software architecture	Content consists of the following topics: - Modelling - Method - Analysis - Design - Project work - IT tools - Experiments - Quality	Content consists of the following topics: - Operating systems - Security - Distributed systems - Networks - Databases - Technological development	Content consists of the following topics: - Organisation and business understanding - Business analysis - IT and business concepts - Financial control - ERP systems - IT purchase - Organisation and IT security

5.0 Compulsory parts of the programme

5.1 First year of study

Aim:

The aim of the first year of study is to give the student competence to work independently and in collaboration with others on the development of primarily single-user systems. Subjects range from analysis to management and operation. The approach will be systematic, involving both technological and business aspects.

5.1.1 Software Construction

The aim of this subject is to qualify the student to create efficient systems with relevant characteristics. The subject is closely related to Software Design and Computer Architecture and Operating Systems.

5.1.2 Software Design

The aim of this subject is to qualify the student for new development, enhancement and integration of IT systems of various types systematically, using a specific modern method and relevant system development tools. The subject qualifies the student to develop IT systems with relevant characteristics efficiently and as such links the other subjects in the 1st year of study together.

5.1.3 Information Technology in Organisations

The aim of this subject is to qualify the student to consider relevant business understanding and company aspects in connection with system development. The subject qualifies the student to work in a system development organisation and take part in the development, enhancement and integration of IT systems for various types of organisation.

5.1.4 Computer Architecture and Operating Systems

The aim of this subject is to qualify the student to assist in the selection and use of technology in connection with system development and programming of single- and multi-user IT systems, and give the student a basic knowledge of technological aspects.

5.2 Second year of study

Aim:

To give the student competence to work independently and in collaboration with others on the evaluation of a company's IT development options and, on this basis, to develop, enhance, and maintain a distributed IT system from analysis to management and operation systematically and methodically in accordance with the given situation.

5.2.1 System Development Methods

The aim of this subject is to qualify the student for the new development, enhancement and integration of distributed IT systems of various types systematically, using modern methods and system development tools relevant to the given situation. The subject qualifies the student to develop appropriate IT systems with relevant characteristics efficiently and as such links the other compulsory subjects in the 2nd year of study together.

5.2.2 Software Architecture and Distributed Programs

The aim of this subject is to qualify the student to create distributed systems with relevant characteristics efficiently. The subject is a core subject closely related to the other compulsory subjects in the 2nd year of study.

5.2.3 Computer Networks and Distributed Systems

The aim of this subject is to qualify the student to assist in the selection and use of technology in connection with the system development and programming of distributed IT systems, and to give the student a comprehensive knowledge of technological aspects.

5.3 Programme structure of each semester

The schedule for the various parts of the programme is shown in the table below and Appendix 11.5

	1st semester	2nd semester	3rd semester	4th sem.	5th sem.
Programming	**Software Construction 25 ECTS** • Programming languages • Database programming • Language theory • Program quality • Algorithms • Templates,techniques and patterns • Data structures and abstract data types • Concurrency		**Software architecture and Distributed Programs 15 ECTS** • Concurrency • Distributed programming • Software architecture • Language theory • Program quality • Algorithms • Data structures and abstract data types, templates, techniques and patterns		
System development	**Software Design 10 ECTS** • Modelling • Method • Analysis • Design • Project work • IT tools • Experiments • Quality		**System Development Methods 15 ECTS** • Method • Design • Project work • IT tools • Experiments • Analysis • Quality		
Technology		**Computer Architecture and Operating Systems 10 ECTS** • Operating systems • Technology development • Databases • Networks	**Computer Networks and Distributed Systems 10 ECTS** • Security • Distributed systems • Networks • Databases		
Business	**Information Technology in Organisations 15 ECTS** • Organisation and business understanding • Business analysis • IT and business concepts • Financial control • ERP systems • IT purchase • Organisation and IT security				
Elective subjects, work experience & main project				20 ECTS	30 ECTS

6.0 Elective: Specialisation (20 ECTS)

The specialisation course gives the student the opportunity to qualify his /her study and business competence by specialising in and putting into perspective topics broadly related to IT.

The specialisation course must have a duration equivalent to 20 ECTS points. The Institute arranges the course by establishing and offering a number of courses within the programme's overall aim. In this connection, both the wishes of the students and the needs of local businesses must be taken into account. The Institute can also make an agreement with the student whereby the specialisation course can take place at another educational institution or with a company.

The student chooses one or more topics for the specialisation course. The topic(s) must go beyond – either in depth or range – the extent they already form as part of the programme.

7.0 Work experience

Work experience (15 ECTS)

In the work experience period, the student will face problems relevant to the curriculum and gain knowledge of relevant business functions. During the work experience period the student will be attached to one or more companies. Work experience can be flexible and differentiated because it must be able to form the basis for the student's final exam project. The work experience period comes after the 1st year of study.

Learning objectives for the work experience period

The aim of work experience with a company is to give the student the opportunity to try out what he/she has learned on the programme so far in practice by functioning in realistic working conditions in a company and job relevant for the profession.

- To gain insight into the requirements and expectations companies have with regard to the knowledge, skills and attitudes of an AP graduate in computer science
- To experience typical life and work in the profession over an extended period
- To work on development projects in practice in accordance with one's own learning goals
- To try out in practice the knowledge and skills gained on the Computer Science programme.
- To gain experience of other working methods and tools for carrying out concrete work tasks

Moreover:

- To get some ideas for the final examination project

Guidelines for the work experience period

The student will have a work experience supervisor from the Institute and a contact person / supervisor from the company.

The student and supervisors / contact person will jointly establish goals for what the student should learn on the basis of the Institute's learning objectives for the work experience period. These goals will then constitute the guidelines for the organisation of the student's work during the period.

The period of work experience has the same requirements with regard to working hours, effort, commitment and flexibility as the AP graduate can expect to face in his/her first full-time job.

The student is entitlted to SU during the work experience period.

8.0 Final examination project (15 ECTS)

In the final examination project (main project) the student must document an ability to work analytically and methodically on a real-life problem that is related to a concrete task in IT. The main project must include central topics in the programme.

Prerequisites

The student must have passed all the previous examinations in order to take the final examination project. Moreover, the work experience period must have been approved.

Content

The research question for the final examination project should be prepared by the student as far as possible in collaboration with a company. The research question must be approved by the educational institution.

In answering the research question, it is important that the student can make use of central theories and methods.

The educational institution will prepare more precise guidelines for the formal project requirements.

Aim:

The aim of the final examination project is that the student should document an ability to work analytically and methodically on a complex and real-life problem that is related to a concrete task in IT. The project has a duration equivalent to 15 ECTS points. The student carries out the final examination project on questions central to the educational programme.

Learning objectives:

Knowledge:

- The student has gained the necessary knowledge, including domain knowledge, to be able to carry out the project.

Skills:

In a practical project, the student can:

- assess and select relevant methods and techniques in relation to the project
- make full use of the methods and techniques used in the project
- plan, manage and carry out the project using relevant methods and techniques
- document his/her results and working process in relation to the requirements of the method(s) used

Competences:

Development competence:

- The student is capable of adapting methods and techniques in relation to the concrete problems faced in the project. Moreover, the student is capable of reflecting over and improving his/her working process.

Teamwork competence:

- The student can engage in qualified dialogue about the project with other specialists and users.

Learning competence:

- The student is capable of understanding new theories, methods and techniques to the extent that is relevant for the project.

9.0 Examinations

1st Semester		2nd semester		3rd semester		4th semester			5th semester		
Software Construction 15 ECTS	INTERNAL EXAM	Computer Architecture & Operating Systems 10 ECTS	1st YEARS EXAM	Computer-Networks & Distributed Systems 10 ECTS	PROGRAMMING EXAM	Specialisa-tion course 20 ECTS	SPECIALISATION EXAM	Work experience 15 ECTS	WORK EXPERIENCE INTERNAL EXAM	Final examina-tion project 15 ECTS	MAIN PROJECT EXAM
Information Technology in Organisations 10 ECTS		10 ECTS		Software, Architecture- and Distributed Programs 15 ECTS							
Software Design 5 ECTS		5 ECTS		System Development Methods 5 ECTS		10 ECTS	SYM INTERNAL EXAM				

9.1 First semester examination (internal)

Basis for the exam:
All the important topics from 1st semester's curriculum in all subjects.
A project of a duration equivalent to 4 ETCS must be handed in.
At least 10 days before the examination, the examination questions for each subject will be made public.

Form of the exam:
Oral examination based on the examination questions. Each exam lasts 20 minutes including giving of marks.
The student draws a question in each subject and starts his/her oral presentation. The examination takes the form of a conversation between the student and the subject's examiner.

Marking:
The subjects have equal weight and the student is marked as either 'Passed' or 'Failed'.

9.2 First year examination (project – external)

The aim of the first year of study is to give the student competence to work independently and in collaboration with others on the development of primarily single-user systems. Subjects range from analysis to management and operation. The approach will be systematic, involving both technological and business aspects.

Learning objectives:

Knowledge:

The student has knowledge about:

- IT requirement assessment
- The selected system development method
- Design
- Implementation (the selected tools, methods, techniques)
- Databases (single-user)

Skills:

- The student can carry out a system development project from idea to running system using IT requirement assessment tools, the selected system development method, the selected language and IDE, and the selected DBMS
- The student can create appropriate and extensive parts of the system using an appropriate architecture
- The student can document the developed system with regard to the selected method with special emphasis on traceability.

Competences:

- The student can reflect over his/her own practice with regard to method, techniques, tools and process.

Basis for the exam:
All four subjects from the first year of study.
The project has a duration equivalent to 12 ECTS and is carried out by project groups on the basis of a description prepared by the school. The project is managed using the methods, techniques and tools used in the teaching. A project report of max. 60 normal pages excluding programs and the executable system on CD-ROM. The report is assessed individually for each student, so it must be clear from the report who is responsible for each part of the project. The whole report is used as the basis in the individual oral examination.

Form of the exam:
Project examination based on a project, including important topics from the first year of study. Each student presents selected parts of his/her project for max. 10 minutes, after which there is an individual examination of 30 minutes including evaluation.

Marking:
An individual overall mark is given on the basis of an overall evaluation of the written and oral parts of the examination. See also Appendix 1 for indicative marking descriptions.

9.3 Programming examination (oral – external)

Learning objectives:

For the programming part see Section 4.1

For the technical part, the following apply:

Knowledge:
The aim is that the student

- knows about the principles for the building of fault-tolerant systems,
- can understand central security concepts , including authorisation, authentication, encryption and logging
- can understand central threats of a technical character that an IT system can face and understand how these threats can be met.
- can understand the principles for the design and creation of distributed systems,
- can understand techniques for the integration of heterogeneous systems.
- can understand the functionality of various types of standard servers, including web servers and application servers
- can understand a stratified communication model
- can understand addressing in networks

- knows about types of net and components in the network.

Skills:
The aim is that the student

- can use mechanisms for synchronisation between threads
- can use standard components for secure communication,
- can use common application protocols in the construction of distributed systems
- can analyse system architectures and can make choices between suggested solutions for a given task.
- can use the services offered by various types of standard servers, including web servers and application servers
- can use a programming interface for communication networks

Competences:
The aim is that the student

- can use basic technological knowledge in connection with system development and programming

Basis for the exam:
The topic Programming and the subject "Computer Networks and Distributed Systems".

Form of the exam:
The student draws a main question in the topic Programming and supplementary questions in the subject Computer Networks and Distributed Systems. The questions cover both theoretical and practical elements. The student prepares answers to the questions for 80 minutes, after which the examination itself lasts 40 minutes including evaluation.

Marking:
The question on programming is weighted at 80% of the overall mark. See also Appendix 2 for indicative marking descriptions.

9.4 System Development examination (project – internal)

Learning objectives:

See Section 4.2

Basis for the exam:
A project of a duration equivalent to 6 ECTS points with fixed aim and framework with emphasis on system development. The basis for the oral part of the exam is the whole topic of System Development. The project is carried out in project groups normally with 3-4 students. A project report of

max. 40 normal pages is handed in. The report is assessed individually for each student, so it must be clear from the report who is responsible for each part of the project. The whole report is used as the basis in the individual oral examination.

Form of exam:
Each student makes a presentation of selected parts of the project of max. 10 minutes, after which there is an individual examination of 30 minutes including evaluation with 40 minutes preparation.

Marking:
An individual overall mark is given on the basis of an overall evaluation of the written and oral parts of the examination. See also Appendix 3 for indicative marking descriptions.

9.5 Specialisation examination (oral – external)

Learning objectives:

Knowledge:
The student must have achieved the following at a level appropriate in relation to the character of the topic(s) selected and the time allowed:

- Knowledge about the selected topic(s) theory and practice.
- Ability to reflect over the relevance of the selected topic(s) to IT's theory and practice.

Skills*:*
The student must master the following skills at a level appropriate in relation to the character of the topic(s) selected and the time allowed:

- The specialist, procedural and analytical skills involved in the selected topic(s).
- Be able to evaluate problems and suggest solution options in relation to the selected topic(s).
- Be able to communicate the main results of the specialisation course – in terms of both product and process –in the written report and the oral presentation.

Competences:
The student must have achieved the following competences:

- Be able to master in new topics in the subject's theory and/or practice.
- Be able to put the selected topic(s) in perspective and relate them to the programme's other topics.
- Be able to reflect over his/her own working and learning process during the specialisation course.

Basis for the exam:
Specialisation subjects.
An obligatory project is handed in for each of the specialisation subjects

Form of the exam:
Course work; the student defines an examination question in one of the specialisation subjects. On the basis of this, a synopsis (10-15 pages) is prepared and handed in before the examination.
The student presents the results of his/her work on the examination question and the synopsis in an oral examination. Moreover, the student draws one question in the curriculum of specialisation subjects. There is a total of 30 minutes for the examination of each student including evaluation, of which approximately 8 minutes are spent on the question on the specialisation subjects.

Marking:
An individual overall mark is given on the basis of an overall evaluation, but the weighting is generally 35% for the synopsis and 65% for the presentation and examination. See also Appendix 7 for indicative marking descriptions.

9.6 Work experience examination (written – internal)

Learning objectives:
The aim of the period of work experience in Computer Science is at give the student the opportunity to qulify his/her study and business competence by specialisation and putting topics broadly related to the aims of the programme in perspective.

The period of work experience comes in the programme's 5th semester, and it links the teaching in the compulsory part with the independent final examination project and is therefore characterised by individual choice and more independent study. The work experience period makes up 15 ECTS.

Knowledge:
The student must:

- Have knowledge about the applied theory, method and practice of the compulsory topics and specialisation subjects
- Be able to understand the concepts and methods in the chosen research question and reflect over their use.

Skills:
The student must:

- Be able to use an all-round set of technical, creative and analytical skills that are linked to employment in the business.
- Be able to evaluate real-life problems and suggestion solution options.
- Be able to communicate real-life problems and suggestion solution options.

Competences:
The student must have achieved the following competences:

- In a structured context be able to master skills and new knowledge related to the business.

Basis for the exam:
The student will have a work-experience superviser from the Institute and a contact person / supervisor from the company. The student and supervisors / contact person will jointly establish goals for what the student should learn on the basis of the Institute's learning objectives for the work experience period. These goals will then constitue the guidelines for the organisation of the student's work during the period.

The work experience period concludes with an evaluation interview based on a written report (max 15 pages), during which the student and supervisor jointly evaluate what the student has learnt from the work experience compared with the targets aimed at. This must be approved before the student can go up to examination.

Up to three studerents can be in work experience in the same function in the same company.

The period of work experience has the same requirements with regard to working hours, effort, commitment and flexibility as the AP graduate can expect to face in his/her first full-time job.

Form of the exam:

The individual student hands in a written report about the selected work experience.

Marking:

The report is marked with either Passed or Failed. See also Appendix 8 for indicative marking descriptions.

9.7 Final examination (main project)

Final examination project is a project. The topic for the final examination project is formulated by the student in consultation with the institution and as far as possible in collaboration with a company. The institution also has to approve the project formulation.

Learning objectives:

The final examination project must demonstrate that the learning objectives and programme's graduation level has been reached.

Basis for the exam

The exam for the final examination project consists in an evaluation of the projects documented results and an oral defence. An individual overall mark is given. The defence is used primarily to ensure that the documented results were prepared by the examinee and secondarily for minor adjustments in the evaluation of the examinee's level.

First, the project's documented results are evaluated by the superviser and external examiner together. Then the student defends the project in front of the superviser and external examiner.

If the final examination project is failed, a revised version of the original project report can be submitted for re-examination.

The examination project is normally carried out in groups of up to three students. The institution decides this in more detail in consultation with the individual students.

The final examination project is handed in to the institution in the form of a report and any product in 3 copies. The report excluding appendices must have a maximum length of 20 pages per student plus 40 pages. The product can for example be a program, a system, an analysis or an investigation. The report is assessed individually for each student, so it must be clear from the report who is responsible for each part of the project. The whole report is used as the basis in the individual oral examination.

The form of the exam

The project that has been prepared is examined though an individual oral defence lasting 30 minutes.

This is made up of an introductory presentation of the project's research question and content lasting a maximum of 10 minutes, after which there is an examination dialogue of approx. 20 minutes.

Marking

An individual overall mark is given on the basis of an evaluation of the report and oral part of the examination. See also Appendix 4.

10.0 Current rules and regulations

This AP programme comes under the following laws and regulations:

• **Lov om erhvervsakademiuddannelser og professionsbacheloruddannelser**: Law no. 207 of 31st March 2008
• **Uddannelsesbekendtgørelsen**: Executive order no. 702 of 3rd July 2009 on
The Business Academy Programme in Information Technology (Computer science AP)
• **Kvalitetsbekendtgørelsen**: Executive order no. 635 of 30th June 2000 on
Quality development and assurance in business academy programmes
• **Adgangsbekendtgørelsen**: Executive order no. 106 of 9th February 2009 on access, enrolment and leave, etc. with regard to some tertiary education programmes
• **Eksamensbekendtgørelsen**: Executive order no. 782 of 17th August 2009 on tests and examinations in business academy programmes
• **Karakterbekendtgørelsen**: Executive order no. 262 of 20th March 2007 on marks and other evaluation
• **Åben Uddannelse**: Law no. 956 of 28th November 2003 on open education, etc.

The laws and executive orders are accessible on the website of the Ministry of Education at www.uvm.dk.

11 .0 Appendices

Appendix 1: Indicative marking descriptions for First Year Examination

Mark	Description	Performance / achievements
12	**For an excellent performance** that demonstrates complete achievement of the examination's goals, with no or very few mistakes/omissions	**Knowledge**: • The student demonstrates exhaustive and certain knowledge about: o IT requirement assessment o The selected system development method o Design o Implementation (the selected tools, methods, techniques) o Database (single-user) **Skills** • The student demonstrates proficiency in carrying out the system development project from idea to running system using IT requirement assessment tools, the selected system development method, the selected language and IDE, and the selected DBMS • The student can create appropriate and extensive parts of the system using an appropriate architecture • The student can document the system developed with regard to the selected method with special emphasis on traceability. **Competences**: • The student can demonstrate reflection over his/her own practice with regard to method, techniques, tools and process. **Examples of mistakes/omissions that still give a mark of 12** • A few minor mistakes or omissions in models or code that do not destroy the whole/the main thread/the traceability of the system.
7	**For a good performance** that demonstrates achievement of the examination's goals with a number of mistakes/omissions	**Knowledge** • The student demonstrates reasonably certain knowledge about central models, their construction and the principles of the method selected • The student demonstrates reasonably certain knowledge about databases. • The student demonstrates reasonably certain knowledge about the programming language selected and the relevant APIs. • The student demonstrates reasonably certain knowledge about IT requirement assessment.

		Skills • The student demonstrates reasonable proficiency in using the IT requirement assessment tools and the system development method selected • The student can create central parts of the system using an appropriate architecture. • The student can to prepare relevant documentation in accordance with the method. **Competences.** The student can reflect over his/her application in terms of method, techniques, tools and process with reasonable proficiency. **Examples of mistakes/omissions that still give a mark of 7** • Minor mistakes/omissions in models or code that do not destroy the whole/the main thread/the traceability of the system.
2	**For an adequate performance**, demonstrating the minimally acceptable degree of achievement.	**Knowledge** • The student demonstrates uncertain knowledge about central models. • The student demonstrates knowledge about the method selected without being able to give details. • The student demonstrates uncertain knowledge about databases. • The student demonstrates uncertain knowledge about the programming language selected and the relevant APIs. • The student demonstrates uncertain knowledge about IT requirement assessment **Skills** • The student demonstrates uncertainty in using the IT requirement assessment tools and the system development method selected • The student can create small parts of the system (minimum: CRUD on two related objects) using an appropriate architecture. • The student can explain the relevant documentation in accordance with the method, but with some uncertainty. **Competences.** • Learning competence: The student is just capable of taking part in the 2nd year of teaching with a reasonable result. **Examples of mistakes/omissions that still give a mark of 2** • Errors in models, code or explanations of them, while retaining a degree of traceability, albeit with uncertainty.

Appendix 2: Indicative marking descriptions for Programming examination

Mark	Description	Performance / achievements
12	**For an excellent performance** that demonstrates complete achievement of the subject's goals, with no or very few mistakes/omissions	**Knowledge** • The student demonstrates comprehensive and certain knowledge about the selected programming techniques • The student demonstrates certain knowledge about fundamental data structures and algorithms • The student demonstrates comprehensive and certain knowledge about the selected programming language and relevant APIs • The student can explain proficiently the central technology related to both centralised and distributed systems **Skills** • The student can select and use fundamental programming techniques with certainty • The student can proficiently use the selected programming language and relevant APIs • The student can proficiently use the selected IDE • The student must be able to demonstrate the ability to create solutions to complex programming tasks **Competences** • Development competence: The student can reflect over the programming practices used, take part in the development and adaptation of programming techniques and methods. • Teamwork competence: The student can engage in dialogue with other specialists about product and process quality • Learning competence: The student can acquire new programming languages with relevant APIs and IDEs. **Examples of mistakes/omissions that still give a mark of 12.** Minor syntax errors, minor mistakes/omissions in explanations of central specialist terms and techniques, lack of knowledge about some details in the subjects.

7	**For a good performance** that demonstrates achievement of the subject's goals with a number of mistakes/omissions	**Knowledge** • The student demonstrates reasonably certain knowledge about the selected programming techniques • The student demonstrates reasonably certain knowledge about fundamental data structures and algorithms • The student demonstrates reasonably certain knowledge about the selected programming language and relevant APIs • The student can explain the central technical requirements for both centralised and distributed systems with reasonable certainty **Skills** • The student can use fundamental programming techniques with reasonable certainty • The student can use the programming language selected and relevant APIs with reasonable certainty • The student can use the IDE selected with reasonable certainty • The student must be able to demonstrate the ability to create solutions for less complex programming tasks • The student can explain the solution options for complex programming tasks with reasonable certainty **Competences** • Development competence: The student must be able to reflect over the programming practice used, and take part in adapting programming techniques. • Teamwork competence: can engage in dialogue with other specialists about product and process quality. • Learning competence: can acquire new programming languages with relevant APIs and IDEs if given guidance. **Examples of mistakes/omissions that still give a mark of 7.** Some uncertainty in knowledge about central specialist terms and techniques, lack of knowledge about some aspects of the subject areas.

2	**For an adequate performance**, demonstrating the minimally acceptable degree of achievement.	**Knowledge** • The student demonstrates uncertain knowledge about fundamental programming techniques • The student demonstrates uncertain knowledge about fundamental data structures and algorithms • The student demonstrates uncertain knowledge about the selected programming language and central parts of the relevant APIs • The student can explain the central technology related to both centralised and distributed systems, but with some uncertainty **Skills** • The student can prepare small, simple programs with reasonable certainty • The student can use fundamental programming techniques with some uncertainty • The student can use the programming language selected and relevant APIs with some uncertainty • The student can use the IDE selected with some uncertainty **Competences** • Development competence: The student can reflect over the programming practice used albeit with some uncertainty. • Teamwork competence: The student can engage in a dialogue with other specialists with some uncertainty. • Learning competence: The student can acquire new programming languages and relevant APIs and IDEs if given guidance. **Examples of mistakes/omissions that still give a mark of 2.** Inability to explain fundamental data structures, but an ability to explain what a data structure is and what data structures can be used for.

Appendix 3: Indicative marking descriptions for System Development examination

Mark	Description	Performance / achievements
12	**For an excellent performance** that demonstrates complete achievement of the subject's goals, with no or very few mistakes/omissions	**Knowledge** The student demonstrates comprehensive and certain knowledge about: • the first-year system development method and at least one other system development method • concepts with which system development methods can be compared • a broad palette of process models • various quality criteria **Skills** The student can with certainty • select a system development method for a given situation on the basis of a systematic comparison • work systematically on a project using the new system development method selected • plan, assess and manage a project using the new system development method selected • relate concrete methods to relevant process models **Competences**. The student can: • adapt a system development method to a project in a given situation • reflect over and compare various methods in practice with certainty • acquire new process models and system development methods **Examples of mistakes/omissions that still give a mark of 12.**
7	**For a good performance** that demonstrates achievement of the subject's goals with a number of mistakes/omissions	**Knowledge** • The student can explain the central principles in the first-year system development method and the principles and practice of at least one other system development method with some certainty • The student knows about a broad range of process models • The student can explain central quality criteria albeit with some uncertainty **Skills** The student can with reasonable certainty:

		• select a system development method appropriate for a given situation • work methodically and systematically with the system development method selected • plan, assess and manage a project using the new system development method selected • relate the system development method selected to relevant process models **Competences** The student can with reasonable certainty: • adapt parts of a system development method in a given situation • reflect over and compare how various methods are used in practice **Examples of mistakes/omissions that still give a mark of 7.**
2	**For an adequate performance**, demonstrating the minimally acceptable degree of achievement.	**Knowledge** • The student can explain the first-year system development method with some certainty • The student can explain the principles and practice of at least one other system development method albeit with some uncertainty • The student knows about some of the principles involved in a systematic comparison of system development methods • The student knows about the fundamental differences between, and the content of, various process models • The student knows about some of the criteria in quality assessment **Skills** The student can albeit with some uncertainty: • take part in development projects that use the system development model selected • plan, assess and manage a project using the new system development method selected **Competences** • The student can take part in adapting a method for a concrete project albeit with some uncertainty • The student can reflect over the selected system development method with some certainty **Examples of mistakes/omissions that still give a mark of 2:**

Appendix 4: Indicative marking descriptions for the Final Examination Project

Mark	Description	Performance / achievements
12	**For an excellent performance** that demonstrates complete achievement of the examination's goals, with no or very few mistakes/omissions	**Knowledge** • The student exhaustively explains in the written report and the oral defence the problems discussed and the theories, methods and techniques used • The student proficiently explains in the written report and the oral defence the project's relationship to relevant parts of the programme's central subjects **Skills** • The student can assess and select relevant methods and techniques with regard to the project with certainty • The student can make full use of the methods and techniques used in the project with certainty • The student can plan, manage and carry out a project using relevant methods and techniques with certainty • The student can document clearly and precisely his/her results and work process with regard to the requirements of the method(s) used **Competences** • Development competence: The student is capable of adapting methods and techniques in relation to the concrete problems faced in the project. Moreover, the student is capable of reflecting over and improving his/her working process. • Teamwork competence: The student can engage in qualified dialogue about the project with other specialists and users. • Learning competence: The student is capable of understanding new theories, methods and techniques to the extent that is relevant for the project. **Examples of mistakes/omissions that still give a mark of 12:** • Minor errors in the report and product that do not seriously affect the whole.

7	**For a good performance** that demonstrates achievement of the examination's goals with a number of mistakes/omissions	**Knowledge** • The student explains in the written report and oral defence with reasonable certainty the problems dealt with and the theories, methods and techniques used • The student explains in the written report and oral defence with reasonable certainty the project's relation to relevant parts of the course's central subjects **Skills** • The student can assess and choose relevant methods and techniques with regard to the project with some certainty • The student uses methods and techniques in the project with reasonable certainty • The student can plan, manage and carry out a project using relevant methods and techniques with reasonable certainty • The student can document his/her results and working process with regard to the requirements of the method(s) used albeit with some mistakes/omissions **Competences** • Development competence: The student is able to adapt methods and techniques with regard to the concrete problems in the project to a certain extent. The student is also able to reflect over his/her working process. • Teamwork competence: The student can engage in dialogue about the project with other specialists and users. • Learning competence: With some guidance, the student is able to understand new theories, methods and techniques to the extent that is relevant for the project. **Examples of mistakes/omissions that still give a mark of 7.** • Some aspects of the problem are not dealt with adequately • Some uncertainty in the use of method and techniques • Some choices are not appropriate or explained • Some lack of documentation

02	**For an adequate performance**, demonstrating the minimally acceptable degree of achievement.	**Knowledge** • The student explains the problems dealt with and the theories, methods and techniques used, albeit with uncertainty and mistakes/omissions in the written report and oral defence • The student explains the project's relationship to relevant parts of the programme's central subjects less than adequately in the written report and oral defence **Skills**. • The student uses methods and techniques in the project, but with uncertainty • The student can only plan, manage and carry out a project using relevant methods and techniques with uncertainty • The student can only document his/her results and working process with uncertainty and with mistakes/omissions **Competences** • Development competence: The student can only adapt methods and techniques with regard to the concrete problems in the project to a limited extent and with guidance • Teamwork competence: The student can engage in dialogue about the project with specialists and users, but with uncertainty • Learning competence: The student is only able to understand new theories, methods and techniques to a limited extent and with guidance **Examples of mistakes/omissions that still give a mark of 2.** • Many choices are inappropriate or unexplained • Methods and techniques are used wrongly and uncertainly • Material mistakes/omissions in the documentation

Appendix 5: Core areas distributed over the three years

Distribution of ECTS points	1st sem.	2nd sem.	3rd sem.	4th sem.	5th sem.	
Programming	15 ECTS	10 ECTS	15 ECTS			40 ECTS
System development	5 ECTS	5 ECTS	5 ECTS	10 ECTS		25 ECTS
Technology		10 ECTS	10 ECTS			20 ECTS
Business	10 ECTS	5 ECTS				15 ECTS
Elective / specialisation course				20 ECTS		20 ECTS
Work experience					15 ECTS	15 ECTS
Final exam project					15 ECTS	15 ECTS
	30 ECTS	30 ECTS	30 ECTS	30 ECTS	30 ECTS	120 ECTS

Appendix 6: Compulsory study elements in the first and second years of study

Subjects / compulsory elements of the programme in the first year of study:	***Software Construction***	***Software Design***	***Computer Architecture and Operating Systems***	***Information Technology in Organisations***
Knowledge:	*The student has knowledge about:* - *criteria for program quality* - *describing formal language syntax and semantics*	*The student has knowledge about:* - *the significance of modelling in connection with system development* - *the significance of experiment as part of or supplement to system development method* - *quality criteria's significance for the system development process and the system's final form*	*The student has knowledge about:* - *facilities in, and construction of, modern operating systems* - *interaction between programming languages and operating systems* - *technological developments, including trends* - *facilities in, and mode of operation of, a modern DBMS* - *the challenges of multi-user systems*	*The student has knowledge about:* - *central organisational concepts* - *e-business and IT strategy* - *IT purchase*
Skills:	*The student can use the selected language and IDE, and the selected DBMS to create small systems with regard to:* - *an appropriate architecture* - *traceability for the other the system development activities* - *quality assurance* - *appropriate use of fundamental algorithms, data structures, patterns*	*The student can use the selected system development method and relevant tools for the development of small database systems from idea to running system with regard to:* - *an appropriate architecture* - *documentation and traceability* - *quality assurance* - *appropriate use of patterns* - *involvement of users* - *user-interface design*	*The student can:* - *use mechanisms for synchronisation of processes and threads*	*The student can take relevant business aspects into account in the development, enhancement, and implementation of IT systems including:* - *carrying out business analysis* - *analysing the organisation of IT security* - *use of ERP systems* - *financial control*
Competences:	***Development competence:*** *The student can reflect over his/her own practice.* ***Teamwork competence:*** *The student can take part in specialist collaboration related to program development.* ***Learning competence:*** *The student can acquire relevant knowledge using the subject area's normal information sources in connection with carrying out of concrete programming tasks.*	*Development competence:* - *The student can reflect over his/her own practice with regard to method and process.* *Teamwork competence:* - *The student can take part in a development project as a competent participant.*	*Learning competence:* - *acquire knowledge about new operating systems and DBMSs*	*Development competence:* - *understands the possibilities in IT for business development* *Teamwork competence:* - *can collaborate with representatives of user and development organisations on the basis of business understanding* - *can take part in project work* *Learning competence:* - *can acquire knowledge about new technology*

Subjects / compulsory elements of the programme in the second year of study:	***System Development Methods***	***Software Architecture and Distributed Programs***	***Computer Networks and Distributed Systems***
Knowledge:	*The student has knowledge about:* - *quality criteria's significance for the system development process and the system's final form* - *the relevance of experiments as a part of, or supplement to, system development methods*	- *The student has knowledge about the qualitative and quantitative properties of classical data structures and algorithms.* - *The student has knowledge about describing formal language syntax and semantics.* - *The student has knowledge about abstraction mechanisms in modern programming languages.*	*The student has knowledge about:* - *central security concepts and threats* - *the principles for design and creation of distributed systems* - *fundamental network concepts*
Skills:	*The student can:* - *select a system development method in a given situation on the basis of a systematic comparison* - *work systematically on a project under the selected new system development method* - *relate concrete methods for relevant process models* - *plan, evaluate and control a small project* - *document and communicate both product and process in connection with system development*	*The student can use the selected language, IDE and relevant APIs to create distributed systems and carry out complex programming jobs with regard to:* - *appropriate selection of fundamental algorithms, data structures and patterns* - *appropriate selection of architecture* - *concurrency problems and interactive processes* - *program quality achieved using relevant tools*	*a student can make use of relevant technological aspects in the development of distributed systems, including:* - *an appropriate system architecture* - *use of a programming interface for communication network* - *use of standard components for secure communication* - *use of common application protocols*
Competences:	*Development competence:* - *adapt a system development method a given project* *Teamwork competence:* - *play a competent role in a development project* *Learning competence:* - *acquire new process models and system development methods* - *reflect over process and method in practice*	*Development competence:* - *the student can take part in the development and adaption of programming techniques and methods.* *Teamwork competence:* - *can engage in dialogue with other specialists about product and process quality.* *Learning competence:* - *the student can reflect over the programming practice used.* - *can acquire new programming languages and relevant APIs and IDEs.*	*Teamwork competence:* - *can collaborate with operating organisation* *Learning competence:* - *acquire knowledge about new technology*

Appendix 7: Indicative marking descriptions for the specialisation examination

Mark	Description	Performance / achievements
12	**For an excellent performance** that demonstrates complete achievement of the examination's goals, with no or very few mistakes/omissions	**Knowledge** The student demonstrates comprehensive and certain knowledge about: • The topic of the specialisation **Skills** The student can with certainty: • Evaluate problems and suggest solution options in relation to the selected topics • Communicate the specialisation's central results – both with regard to product and process. **Competences** The student can: • Independently find out about new topics in the specialisation's theory and/or practice • Put the topics into perspective and relate them to the programme's other topics **Examples of mistakes/omissions that still give a mark of 12:** • Minor mistakes in the synopsis and presentation the do not seriously affect the whole.
7	**For a good performance** that demonstrates achievement of the examination's goals with a number of mistakes/omissions	**Knowledge** The student can explain with reasonable certainty: • The topic of the specialisation **Skills** The student can with reasonable certainty: • Evaluate problems and suggest solution options in relation to the selected topics • Communicate the specialisation's central results – both with regard to product and process. **Competences** The student can with reasonable certainty: • Independently find out about new topics in the specialisation's theory and/or practice • Put the topics into perspective and relate them to the programme's other topics

		Examples of mistakes/omissions that still give a mark of 7: • Individual topics in the synopsis not dealt with adequately • Some uncertainty in the use of new knowledge • Some choices are not appropriate or explained

02	**For an adequate performance**, demonstrating the minimally acceptable degree of achievement.	**Knowledge** The student can with some certainty explain: • The topic of the specialisation **Skills** The student can with some certainty: • Evaluate problems and suggest solution options in relation to the selected topics • Communicate some of the specialisation's central results – both with regard to product and process. **Competences** The student can with some certainty: • Independently find out about new topics in the specialisation's theory and/or practice • Put the topics into perspective and relate them to the programme's other topics **Examples of mistakes/omissions that still give a mark of 2:** • Many choices are not appropriate or explained • Quite a lot of knowledge is used incorrectly or uncertainly. • Quite a few important mistakes/omissions in the synopsis.

Appendix 8: Indicative marking descriptions for the Work Experience examination

Mark	Description	Performance / achievements
Passed	**For an adequate performance**, demonstrating the minimally acceptable degree of achievement.	**Knowledge** The student can with some certainty: • Explain the theory and method behind the practice **Skills** The student can with some certainty: • Use technical, creative and analytical skills in relation to the work practice • Evaluate practical problems and suggest solution options • Communicate practical problems and solution options **Competences** The student can with some certainty: • Acquire new knowledge and skills in relation the work practice. **Examples of mistakes/omissions that still give a mark of Passed:** • Quite a few important mistakes or omissions in the report.