



Curriculum part 4 - 2014

Course Descriptions AP Degree in Computer Science

Datamatikeruddannelsen

Version 1.3
Revised August 2015

Table of contents part 4 – Course Descriptions

1.	Course Description for Programming (PRO)	2
2.	Course Description for Business and Information Technology (BIT)	4
3.	Course Description for Software Development (SD)	6
4.	Course Description for Databases and Operating Systems (DAOS).....	8
5.	Course Description for Distributed Programming (DIP)	10

1. Course Description for Programming (PRO)

Introduction

This course description provides a brief description of the Programming course (PRO1 and PRO2) in the 1st and 2nd semesters of the Computer Science programme at Business Academy Aarhus.

Contents

This course is to qualify the student to develop the competences required to implement IT systems with a user interface. The course also teaches the student how to master the fundamental elements of computer science in the perspective of workmanship.

The course spends most of the time teaching the student to program. The student is taught programming and how to use the object-oriented programming language, Java.

The course also teaches the student program solving processes and program design – the art of developing good programs.

The Programming course forms an integral part with the Software Development course (SD) thereby opening up for cross-disciplinary use of tools and techniques. More specifically, the models developed in Software Development form the basis of programs developed in Programming.

ECTS points

27 ECTS points (15 ECTS in the 1st semester and 12 ECTS in the 2nd semester).

1 ECTS point is equivalent to 10 lessons or 2½ standard days of study.

Learning objectives

Knowledge and understanding

The student will gain knowledge about:

- Definition of abstract data types
- Criteria for program quality
- Abstraction mechanisms in modern programming language

Skills

The student will get the skills to:

- Specify and construct algorithms
- Use the programming language to realise algorithms, design patterns, abstract data types, data structures, design models and user interfaces
- Apply state-of-the-art integrated developments tools, including a version control system
- Develop applications based on a layered software architecture
- Use software components/libraries
- Produce documentation in accordance with current de facto industry standards
- Apply modern techniques and tools to carry out testing and quality assurance
- Assess the qualitative and quantitative properties of algorithms and data structures

Competencies

The student will learn to:

- Be part of development/maintenance projects as a professional programmer
- Learn new skills of relevance to programming languages, development tools, programming techniques and program design

The course comprises the following topics in order to teach the student the knowledge, skills and competences specified above.

Topics 1st semester

- Fundamental programming concepts: types, selection, iteration, methods, parameters, encapsulation
- Objects and classes
- Realisation of the relationship between classes (association, aggregation, composition and specialisation)
- Development of simple graphic user interfaces
- Development of a simple, layered architecture
- Use of standard classes for program realisation
- Realisation of simple algorithms: searching, sorting etc.

Topics 2nd semester

- Use and realisation of recursion (breaking up, solving and combining, advanced sorting algorithms)
- Use and realisation of design patterns
- Realisation of abstract data types (stacks, queues etc.) and data structures (lists, trees etc.)
- Implementation of unit testing.
- Use of version control systems
- Use of GUI builder for realisation of graphic user interfaces.

Tuition

The course will apply various forms of teaching. There will be lessons in class, where new aspects will be taught, and there will be a considerable share of project work where the students work in project groups. There will also be assignments that the students are to work on two and two and in groups. The course involves a considerable degree of hands-on work with assignments.

A course schedule covering the entire semester can be found on frontier. This describes the topics taught in the respective lessons. As the course progresses, additional details about topics, literature and assignments of relevance to the lessons will be published on frontier.

Tools

We use Java as the programming language and Eclipse as the development environment.

Study activity level and attendance

Throughout the semester the students will be asked to work on different types of assignments.

Projects: Large, cross-disciplinary assignments to be addressed in project groups. All students in a group are expected to dedicate their efforts to the project at hand and make sure that all the other students in the group benefit from the work involved. To round off the project the students make a presentation of the outcome of the project.

Exercises: Exercises will typically be given during lessons and they serve to explain and clarify the theory discussed and make it more concrete. Dedicated involvement in the problems introduced in the assignments is a condition for optimum benefit of the instruction. Selected exercises will be discussed in class.

Assignments: There will be assignments to be submitted during the course. These must be submitted on time.

Exam project: The 2nd semester comprises a multidisciplinary project involving all 2nd semester courses. The students work with the project in groups of maximum 3 students. This project is part of the 1st semester test.

The curriculum defines the **minimum** study activities that a student has to participate in to fulfil our requirements to study activity level and attendance. Some of these study activities form part of PRO and are as follows,

Study activities – 1st semester

- PRO1 – four activities
- One joint project with SD1
- 1st semester test

Study activities – 2nd semester

- PRO1 – one assignment to be submitted
- One joint project with SD2

Examination

All students sit for an in-house test at the end of the 1st semester. This test concerns all three 1st semester courses – BIT1, SD1 and PRO1 – and the students will be awarded a Pass or a Fail.

The final course examination of the 1st year test takes place at the end of the 2nd semester. This exam is described in the curriculum.

2. Course Description for Business and Information Technology (BIT)

Introduction

The course description offers a brief description of the Business and Information Technology (BIT) course.

Contents

The objective of this core area is to qualify the student to develop the competences required to include relevant company aspects as well as business understanding in connection with system development.

The core area also qualifies the student to develop the competences required to work in a system development organisation and to participate in development, enhancement and integration of IT systems for different types of organisations.

ECTS points

10 ECTS

1 ECTS point is equivalent to 10 lessons or 2½ standard days of study.

Learning objectives

Knowledge and understanding

The student will gain knowledge about:

- How IT can enhance business processes and develop business activities
- Standard business systems, including organisational concepts

- The rationale of IT investments
- IT security

Skills

The student will get the skills to:

- Analyse and model business processes
- Participate in project work
- Apply innovative approaches focusing on project work in practice-related development projects
- Communicate with internal and external stakeholders and make presentations
- Participate in IT implementation processes and change management processes

Competencies

The student will learn to:

- Participate in and identify the relationship between design of business processes and design of IT systems
- Cooperate with representatives from the user organisation and the development organisation based on business understanding
- Acquire knowledge about new technologies in a business context

The course comprises the following topics in order to teach the student the knowledge, skills and competences specified above.

Topics 1st semester

- Organisational and business understanding
- Information technology and business concepts
- E-business
- Business processes

Topics 2nd semester

- Innovation and change management
- Communication and presentation
- IT security

Tuition

A typical course day involves changing between theory, exercises and review of these. The activities of specific course days will be detailed in a course schedule. The students are required to study considerable quantities of literature.

Litteratur

Skriver, Staunstrup, Henningsen: Organisation, Trojka 2012, 5th ed.
Additional notes and photocopies will be handed out.

Study activity level and attendance

The students will be given assignments to work on from one lesson to the next. The assignments serve to explain and clarify the theory taught and render it more concrete. Dedicated involvement in the problems introduced in the assignments is a condition for optimum benefit of the instruction.

The curriculum defines the **minimum** study activities that a student has to participate in to fulfil our requirements to study activity level and attendance. Some of these study activities form part of BIT1 and are as follows,

Study activities – 1st semester

- BIT 1 – one assignment to be submitted
- 1st semester test.

Study activities – 2nd semester

- BIT 2 – one assignment to be submitted
- Multidisciplinary exam project comprising all 2nd semester courses

Examination

All students sit for an in-house test at the end of the 1st semester. This test concerns all three 1st semester courses – BIT1, SD1 and PRO1 – and the students will be awarded a Pass or a Fail.

The final course examination of the 1st year test takes place at the end of the 2nd semester. This exam is described in the curriculum.

3. Course Description for Software Development (SD)

Introduction

This course description provides a brief description of the Software Development course (SD1 and SD2) in the 1st and 2nd semesters of the Computer Science programme at Business Academy Aarhus.

Contents

The Software Development course provides the student with the competences required to take part in the development of IT systems at a professional level, i.e. contribute actively to processes and apply methods and techniques to analyse a problem. Using methods, tools and techniques the student learns how to set up models incorporating the requirements specified to a new system. These requirements may concern the targeted objective(s), the contents, the design of a system etc. Software Development also teaches the student how to design a system that lives up to specific quality criteria. The student also learns how to define tests intended to verify the quality in a structured manner.

ECTS points

15 ECTS points (9 ECTS in the 1st semester and 6 ECTS in the 2nd semester).

1 ECTS point is equivalent to 10 lessons or 2½ standard days of study.

Learning objectives

Knowledge and understanding

The student will gain knowledge about:

- The importance of experiments as part of or as supplement to the software development method
- The importance of quality criteria for the software development process and the final design of the system

Skills

The student will get the skills to:

- Model and design IT systems
- Apply an appropriate software architecture
- Document and communicate product and process, including ensure traceability
- Run quality assurance procedure on product and process
- Apply appropriate design patterns
- Incorporate users
- Design user interfaces

Competencies

The student will learn to:

- Participate as a competent partner in a development project
- Reflect on and adjust process and method in practice

The course comprises the following topics in order to teach the student with the knowledge, skills and competences specified above.

Topics 1st semester

- Produce UML models using a tool (UML is an international standard diagramming language)
- Define requirements and find and describe use cases
- Find and describe classes and objectives as system elements
- Define design criteria as basis for the chosen architecture etc.
- Understand and choose appropriate architectures

Topics 2nd semester

- Produce more UML models using a tool (UML is an international standard diagramming language)
- Apply GRASP patterns in the system design process
- Prepare and run tests
- Plan and carry out experiments as a supplement to other software development methods
- Apply the Unified Process software development method
- Be able to understand, define and work with usability and incorporate this in the user interface design
- Document process and product as part of the software development process

Tuition

The Software Development course teaches the student how to produce a range of models using the international standard diagramming language, Unified Modelling Language (UML). As part of the teaching, the theory of the different models will be presented and examples shown. As a next step the students will be given assignments where they have to build their own models. Since the Software Development (SD) course is linked closely to the Programming (PRO) course, the students will be programming some of the models developed during the course. During the 1st and 2nd semesters, both courses involve several cross-disciplinary projects where the relationship between the models developed in the SD course and programs developed in the PRO course is made even clearer.

A course schedule covering the entire semester can be found on frontier. This describes the topics taught in the respective lessons. As the course progresses, additional details about topics, literature and assignments of relevance to the lessons will be published on frontier.

Tools

In the Software Development course students are to analyse and design small system using the 'Visual Paradigm Community Edition' CASE tool.

Study activity level and attendance

The students will be given assignments to work on from one lesson to the next. The assignments serve to explain and clarify the theory taught and render it more concrete. Dedicated involvement in the problems introduced in the assignments is a condition for optimum benefit of the instruction.

The curriculum defines the **minimum** study activities that a student has to participate in to fulfil our requirements to study activity level and attendance. Some of these study activities form part of SD1 and are as follows,

Study activities – 1st semester

- SD1 – one assignment to be submitted
- One joint project with PRO1
- 1st semester test

Study activities – 2nd semester

- SD2 – 1 assignment to be submitted
- One joint project with PRO2
- Multidisciplinary exam project comprising all 2nd semester courses

In addition to this additional study activities might be offered.

Examination

All students sit for an in-house test at the end of the 1st semester. This test concerns all three 1st semester courses – BIT1, SD1 and PRO1 – and the students will be awarded a Pass or a Fail.

The final course examination of the 1st year test takes place at the end of the 2nd semester. This exam is described in the curriculum.

4. Course Description for Databases and Operating Systems (DAOS)

Introduction

This is a short description of the subject Databases and Operating Systems (DAOS), taught at the 2nd semester of the Computer Science Programme at Business Academy Aarhus.

Contents

The purpose of the subject is for the student to acquire knowledge of technology and the ability to combine this knowledge with programming skills.

The teaching therefore places great emphasis on the ability to combine technology and programming, both in the theoretical content and in the subject's assignments.

As indicated by the subject title, the focus is on the technical aspects of operating systems and databases. In this subject, the emphasis is on what happens inside one computer. However, the

teaching stresses that what the students learn is equally applicable to large systems involving several computers.

The subject is followed in the 3rd semester by the subject Distribution, Integration and Security, which focuses on the problems that arise when computers are used by multiple users. The two subjects are therefore closely related.

ECTS points

8 ECTS points (during the 2nd semester).

1 ECTS point is equivalent to 10 lessons or 2½ standard days of study.

Learning objectives

Knowledge and understanding

The student will gain knowledge about:

- The facilities and design of modern operating systems
- The facilities and mode of operation of modern database systems
- Problems relating to multiple users

Skills

The student will get the skills to:

- Apply mechanisms for synchronising processes and threads
- Implement models in a database system and create programs that use a database interface
- Design and create programs in the form of interrelated processes/threads

Competencies

The student will learn to:

- Acquire knowledge about new operating and database systems

The course comprises the following topics in order to teach the student with the knowledge, skills and competences specified above.

Topics

- Principles involved in relational databases and how they relate to logical models
- The SQL query language
- Using a programming language to access databases (JDBC)
- Views, stored procedures and triggers
- The mode of operation of a modern DBMS
- The modern operating system, including capabilities, structure and operation
- Processes and threads
- Programming of threads, including problems relating to synchronisation and deadlock
- Disks and files

Tuition

The course comprises different forms of teaching, including classroom lessons explaining new subject matters and a fair amount of assignments. The assignments may be individual or group-based. The teaching therefore involves a lot of practical work with assignments, either individual or group assignments.

A study plan for the entire semester is available on Fronter, where the topics covered in the different lessons are described. More detailed information on the topics, literature and assignments covered in the lessons will be posted on Fronter on a regular basis.

Tools

We use the Java programming language, the Eclipse development environment, and Microsoft SQL Server is the DBMS used.

Study activity level and attendance

Different types of assignments will be given during the semester.

Group projects: A somewhat large assignment to be answered in a project group. All members of the group are expected to work diligently with the project to ensure that all members of the group benefit from the assignment. The group project is to be handed in.

Practical exercises: Typically completed during lessons and consisting of assignments that illustrate, study and exemplify the theory taught. Dedicated work with the problems introduced in the assignments is required if the student is to derive maximum benefit from the teaching. The exercises are dealt with in class.

Assignment to be submitted: An assignment to be submitted is handed out during the course.

Exam project: During the 2nd semester, the students complete a project covering all 2nd semester subjects. The project is completed in groups of max three participants. This project is part of the first-year exam.

The curriculum states which study activities constitute the **minimum** required to meet the demands for study activity and participation. Some of these study activities can be found in DAOS and include the following:

Study activities

- Submission of the result of the group project
- Submission of an individual assignment
- Both must be uploaded in time to Fronter

Examination

The subject is included in the 1st year examination, which takes place after the 2nd semester. A description of this exam is included in the curriculum.

5. Course Description for Distributed Programming (DIP)

Introduction

This is a short description of the subject Distributed Programming (DIP) on the 3rd semester of the Computer Science Programme at Business Academy Aarhus.

Contents

The purpose of the subject is for the student to acquire the ability to master more advanced elements of computer science in the perspective of workmanship and to give the student the ability to implement distributed software systems.

The subject Distributed Programming collaborates with the subject Systems Development (SD). The techniques and tools of the subject are to be utilized in implementing a project in SD as a multi-user web-based project with joint data. More precisely, JSF must be used on the server-side and JavaScript on the client-side – and the other technologies of the subject.

ECTS points

10 ECTS points.

1 ECTS point is equivalent to 10 lessons or 2½ standard days of study.

Learning objectives

Knowledge and understanding

The student will gain knowledge about:

- Integration between heterogeneous components and platforms.

Skills

The student will get the skills to:

- Use techniques for the construction of computer programs with several concurrent users.
- Design and construct computer programs based on co-operating processes in a distributed architecture.
- Construct computer programs which use the latest network technologies.
- Use design patterns for distributed software architecture.
- Develop software components.
- Develop web applications.

Competencies

The student will learn to:

- Participate as a professional programmer in development- and integration projects.
- Acquire new skills in the area of computer languages, development tools, programming techniques and program design.

The course comprises the following topics in order to teach the student with the knowledge, skills and competences specified above.

Topics

- HTML og XHTML
- CSS
- XML
- Simple server
- JAXB
- Webservice
- JavaScript
- DOM
- JQuery
- Regex
- Serializable
- JavaBeans
- JSF
- MVC
- Implementing the SDM project

Tuition

The course comprises different forms of teaching, including classroom lessons explaining new subject matters and a fair amount of assignments. A course schedule covering the entire semester can be found on Fronter. More detailed information on the topics, literature and assignments covered in the lessons will be posted on Fronter on a regular basis.

Tools

We use Java and JavaScript and a modern development environment.

Study activity level and attendance

Different types of assignments will be given during the semester.

Exercises: Exercises will typically be given during lessons and they serve to explain and clarify the theory discussed and make it more concrete. Dedicated work with the problems introduced in the assignments is required if the student is to derive maximum benefit from the teaching. Selected exercises are dealt with in class.

SDM project: Be able to use the techniques and tools defined to implement the SDM project.

The curriculum states which study activities constitute the **minimum** required to meet the demands for study activity and participation. Some of these study activities can be found in DIP and include the following:

- Programming of the client-side in the SDM project with JavaScript.
- Programming of the server-side in the SDM project with JSF.

Examination

The student is examined after 3rd semester in the programming exam. This exam is described in the curriculum.