

Curriculum for the
Bachelor's Degree
Programme in
Software Development
National section



1. Contents

1.	Contents.....	0
2.	The programme’s goals for learning outcomes	1
3.	The programme includes four national subject elements	2
3.1	Developing Large Systems.....	2
3.2	Databases for Developers	3
3.3	System Integration	4
3.4	Testing	5
3.5	The number of exams in the national subject elements.....	5
4.	Internship.....	6
5.	Requirements for the bachelor project	6
6.	Rules on credit.....	7
6.1	Academic criteria for selecting candidates for top-up programmes.....	7
7.	Commencement and transitional schemes	7

In the event of any discrepancy between this translation and the Danish original, the Danish text shall prevail.

This national part of the curriculum for the Bachelor’s Degree Programme in Software Development has been released in accordance with section 18(1) in the Ministerial Order for technical and commercial Academy Profession Programmes and Professional Bachelor Programmes. This curriculum is supplemented with an institutional component, provided by the institution offering the programme.

After it has been approved by either the Board of Directors (or the Rectors) and after consultation with the institutions’ Educational Committee and the External Examiners’ chairmanship for the specific programme, the educational network for the Bachelor’s Degree Programme in Software Development prepares the institutional part.

2. The programme's goals for learning outcomes

The goals for learning outcomes cover the knowledge, skills and competencies that a professional bachelor in software development must achieve during the programme.

Knowledge

The student must have knowledge of:

- The strategic role of testing in system development
- Globalisation of software production
- System architecture and its strategic importance for the company's business
- Applied theory and methodology and common technologies within the domain
- Various database types and their applications.

Skills

The student can:

- Integrate IT systems and develop systems that support future integration
- Use contracts as a control and coordination mechanism in the development process
- Assess and select database systems, and design, redesign and optimise databases
- Plan and manage development processes involving many geographically separated project participants
- Identify links between applied theory, methods and technology and reflect on their suitability in various situations

Competencies

The student can:

- plan and implement testing of large IT systems
- Engage in professional collaboration to develop large systems by applying common methods and technologies
- Familiarise themselves with new technologies and standards for handling integration between systems,
- Through practice, develop their own competency profile from a primarily back-end developer profile to performing tasks as a system architect
- Handle the establishment and realisation of both a business and technologically appropriate architecture for large systems.

3. The programme includes four national subject elements

3.1 Developing Large Systems

Developing Large Systems
Scope: 10 ECTS
Content: The contents of the subject element are design, implementation and maintenance of large, distributed IT systems, where scalability is a central issue. Focus is on system development methods, techniques and technologies for continuous integration and delivery. In addition, development in distributed developer teams is considered.
Learning objectives: Knowledge The student must have knowledge of: <ul style="list-style-type: none">• Issues related to the development of distributed and large-scale IT systems, and how disciplined and agile development methods prescribe how these issues should be handled• The advantages, disadvantages and costs of using a system for the continuous integration and delivery of IT systems• Quality criteria for the design of interfaces to subsystems.• Configuration- and error reporting systems dedicated to development of large, distributed systems Skills The student can: <ul style="list-style-type: none">• Apply techniques for dividing a system into subsystems• Design and specify requirements for subsystems• Use version control systems dedicated to the development of large distributed systems in a distributed development team• Use a system for continuous integration and delivery• Use architecture patterns dedicated to the development of large distributed systems. Competencies The student can: <ul style="list-style-type: none">• Develop large distributed systems in a distributed development team• Adapt development methods and processes to the development of large distributed systems

3.2 Databases for Developers

Databases for Developers
Scope: 10 ECTS
Content: The contents of the subject element are various database types, selection and use of various database types and their application in different domains. Furthermore, analysis and development, including re-design and optimisation of large databases are included.
Learning objectives: Knowledge The student must have knowledge of: <ul style="list-style-type: none">• Various database types and the underlying models• A specific database system's storage organisation and query execution• A specific database system's optimisation possibilities – including advantages and disadvantages• Database-specific security problems and their solutions• Concepts and issues in relation to data warehousing, including big data• The particular issues raised by having many simultaneous transactions, including in connection with distributed databases• Relational algebra (including its relationship to execution plans) Skills The student can: <ul style="list-style-type: none">• Transform conceptual data models into physical models in various database types• Implement database optimisation• Use parts of the administration tool to assist in the optimisation and tuning of existing databases, including the incorporation of a specific DBMS' execution plans• Use a specific database system's tools for handling simultaneous transactions• Use the programming and other facilities provided by a modern DBMS• Use an object-relational mapping tool Competencies The student can: <ul style="list-style-type: none">• Analyse the application domain in order to select a database type• Divide responsibility for tasks between the application and DBMS during system development, to ensure the best possible implementation.

3.3 System Integration

System Integration

Scope: 10 ECTS

Content: The subject element covers technical systems integration, including integration of existing systems, integration of existing systems into development of new systems and development of new systems that support future integration.

Learning objectives:

Knowledge

The student must have knowledge of:

- Business considerations in relation to system integration
- Standards and standards organisations
- Storage, transformation and integration of data sources
- The concept of services and their relations to service-oriented architectures
- Technologies which can be used to implement a service-oriented architecture
- Tools for integration.

Skills

The student can:

- Use an object-oriented system in a service-oriented architecture
- Design a system that is easy to integrate with other systems, and uses existing services
- Transform or expand a system so that it can function in a service-oriented architecture
- Use patterns that support system integration
- Integrate generic and other systems
- Choose from various integration methods
- Translate elements in a business strategy into specific requirements for system integration.

Competencies

The student can:

- Choose from various integration techniques
- Acquire knowledge of developments in standards for integration
- Adapting of IT architecture to take into account future system integration.

3.4 Testing

Testing
Scope: 10 ECTS
Content: The contents of the subject element are planning and conducting of software tests. This includes the placing and significance of testing in system development methods. Focus is on systematic design and execution of tests of large systems, including automatic testing. Furthermore, techniques for designing and constructing testable systems are covered.
<p>Learning objectives:</p> <p>Knowledge The student must have knowledge of:</p> <ul style="list-style-type: none"> • Significant test strategies and models and their role in system development • Testing as an integral part of a development project • Various types of testing and their applications. <p>Skills The student can:</p> <ul style="list-style-type: none"> • Ensure traceability between system requirements and testing at all levels • Apply both black-box and white-box testing techniques • Apply various criteria for the degree of test coverage • Use techniques for verification and validation • Use techniques and tools for automated testing • Build systems to manage testing and the debugging process in development projects. <p>Competencies The student can:</p> <ul style="list-style-type: none"> • Define, plan and carry out testing in a development project that matches the project's quality requirements • Plan and manage the implementation of internal and external testing of software systems. • Design testable systems

3.5 The number of exams in the national subject elements

There are 4 exams in the national subject elements, as well as one further exam in the bachelor project. For the number of exams in the internship, please refer to section 3.

For a comprehensive overview of all the programme's exams, please refer to the institutional part of the curriculum, as the national subject elements described in this curriculum can be examined together with the subject elements specified in the institutional part of the curriculum.

4. Internship

Learning objectives for the programme's internship

Internship
Scope: 15 ECTS
Content: The internship is organised in cooperation with student, such that it – in combination with the rest of the study programme– provides possibilities for working with concrete and practical tasks in the field of software development. This includes application of theories, methods and techniques.
<p>Learning objectives:</p> <p>Knowledge The student must have knowledge of:</p> <ul style="list-style-type: none"> • Daily operations throughout the internship company. <p>Skills The student can:</p> <ul style="list-style-type: none"> • Apply versatile technical and analytical working methods linked to employment within the profession • Evaluate practice-oriented issues and identify possible solutions • Manage the structuring and planning of day-to-tasks within the profession • Communicate practice-oriented issues and reasoned solution proposals. <p>Competencies The student can:</p> <ul style="list-style-type: none"> • Handle development-oriented, practical and professional situations in relation to the profession. • Acquire new knowledge, skills and competencies related to the profession • Participate in academic and interdisciplinary collaboration with a professional approach.
Number of examinations: 1

5. Requirements for the bachelor project

The learning objectives for the bachelor project are identical to the programme's learning objectives listed above under section 1.

Requirements to the Bachelor's project

The bachelor's project must document the student's understanding of and ability to reflect on the practices of the profession and the use of theory and methods in relation to a real-life problem. The student, possibly in collaboration with a private or public company, formulates the problem statement, which must be central to the programme and profession. The Academy approves the problem statement.

Assessment:

The Bachelor project completes the programme on the last semester when all previous examinations have been passed.

ECTS credits:

The Bachelor project credits 15 ECTS points.

Assessment requirements

The exam is an oral and written examination with an external co-examiner. A combined mark is given based on the 7-point scale for the written project and the oral presentation and examination.

6. Rules on credit

Passed programme elements are equivalent to similar programme elements taken at other educational institutions offering this programme.

Students are obliged to inform the institution of any completed educational elements from another Danish or foreign higher education programme or any jobs which are likely to provide credit. The Academy approves credit, in each instance, based on completed programme elements and any jobs, which meet the objectives of the subjects, the educational part and the internship parts. The decision is based on an academic assessment.

For prior credit approval of studies in Denmark or abroad, students are required to document each approved and completed programme element on the completion of these studies. In connection with applying for prior credit approval, the students give the Academy permission to obtain the necessary information after the student's completion. Following approval according to the above, the programme element is deemed to be passed if it was passed according to the rules of the programme in question.

6.1 Academic criteria for selecting candidates for top-up programmes

If the AP Computer Science programme is completed, then the formal admission requirements for the Bachelor's Degree in software development are met.

If, for reasons of capacity, not all applicants are admitted to the programme, the applicants will be prioritized by the following criteria's:

Applications are assessed based on the following criteria:

- Average grade from the qualifying study programme
- Grades and ECTS credits in programming and system development
- Relevant work experience

7. Commencement and transitional schemes

Commencement

All enrolled students will be transferred to this curriculum on 1 September 2017.

Simultaneously, previous joint national curricula are NOT valid from this date.