COURSE SYLLABUS

Kompilator- och översättarteknik

Compiler Design and Translation Techniques

7,5 ECTS credit points (7,5 högskolepoäng)

1 Course title and credit points
The course is titled Compiler Design and Translation Techniques/Kompilator- och översättarteknik and awards 7,5 ECTS credits. One credit point (högskolepoäng) corresponds to one credit point in the European Credit Transfer System (ECTS).

2 Decision and approval
This course is established by Dean 2016-09-26. The course syllabus was revised by Head of Department of Computer Science and Engineering and applies from 2017-01-16. BTH-4.1.1-0578-2016

3 Objectives
Everybody working within computer science are major users of compilers and translators. Thus it is of great importance to know their function, partly in order to be able to deem their quality, partly in order be an advanced user. Furthermore, one is often in need of a simple or even more complicated translator in which case one may need to personally design one. The technique used in translation and compiling is also applicable to many other ares, whereas knowledge of this subject is particularly useful.

4 Content
The course comprises the following:
• Basic theory for formal languages, grammar and translation
• Principles for lexical, syntactic and semantic analysis
• Code generation
• Execution structures
• Tools for compiler design, orientation
• Compiler design

5 Aims and learning outcomes
Knowledge and understanding.

On completion of the course the student will understand and employ:
• Basic theory for formal languages, grammar and translation
• Principles for lexical, syntactic and semantic analysis

Competence and skills
On completion of the course the student will design, implement and analyse:
• Lexing and parsing for a language
• Code generation for a language
• Test suite that ensures language coverage

Judgement and approach
On completion of the course the student will:
• Assess the impact of choices in representation and algorithms on the implementation of a compiler.

6 Learning and teaching
The course uses a mixture of lectures, supervised lab sessions and project work. The lectures introduce the theoretical material, the labs are used initially for supplied exercises and then for tutorial support on the project. The project is split into two parts to allow extensive feedback early in the process. The teaching language is Swedish.

7 Assessment and grading
Examination of the course

<table>
<thead>
<tr>
<th>Code Module</th>
<th>Credit</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1705 Project part 1</td>
<td>3.5 ECTS</td>
<td>A–F</td>
</tr>
<tr>
<td>1715 Project part 2</td>
<td>4 ECTS</td>
<td>A–F</td>
</tr>
</tbody>
</table>

The course will be graded A Excellent, B Very good, C Good, D Satisfactory, E Sufficient, FX Insufficient, supplementation required, F Fail. The final grade on the course is determined by a weighted and rounded off average of the grades on the two subjects examined. If the weighted grade is exactly in between two grades, it is rounded off to the lower grade.
8 Course evaluation
The course coordinator is responsible for systematically gathering feedback from the students in course evaluations and making sure that the results of these feedback into the development of the course.

9 Prerequisites
The student must have successfully completed 22.5 credits in programming include Algorithm and data structures.

10 Field of education and subject area
The course is part of the field of education and is included in the subject area Computer Science.

11 Restrictions regarding degree
The course cannot form part of a degree with another course, the content of which completely or partly corresponds with the contents of this course.

12 Additional information
Replaces DV1465

13 Course literature and other teaching material
Compilers Principles, Techniques & Tools, international edition
Publisher: Addison-Wesley Publ. BV
Published: 2007
ISBN10: 0321491696
ISBN13: 9780321491695