Course Title: Compiler Construction

Course code: 750421

Course prerequisite(s) and/or corequisite(s): 751323 + 750321

Credit hours: 3

Course Description:
This module introduces topics include compiler design, lexical analysis, parsing, symbol tables, declaration and storage management, code generation, and optimization techniques.

Course Objectives:
The aim of this module is to show how to apply the theory of language translation introduced in the prerequisite courses to build compilers and interpreters. It covers the building of translators both from scratch and using compiler generators. In the process, the module also identifies and explores the main issues of the design of translators.
The construction of a compiler/interpreter for a small language is a necessary component of this module, so students can obtain the necessary skills.

Course Components:
- Introduction to Compilers
- Lexical Analysis
- Syntax Analysis
- Parsers Implementation
- Semantic Analysis
- Intermediate Representation, code generation
- Code generation and Code optimization
- Error Detection and Recovery
- Error Repair, Compiler Implementation
• Compiler design options and examples: C Compilers
• C++, Java, and YACC Compilers

Text book:
Title: Compilers Principles, Techniques and Tools
Author(s)/Editor(s): Alfred V. Aho, Ravi Sethi and Jeffry D. Ulman
Publisher: Addison Wesley Longman, 1986
ISBN: 0-201-10088-6

In addition to the above, the students will be provided with handouts by the lecturer.

Teaching Methods:
Duration: 16 weeks, 48 hours in total
Lectures: 32 hours (2 per week)
Tutorials: 14 hours, 1 per week (except the last two weeks)
Seminars: 2 hours, (1 per week for the last two weeks)

Laboratory projects
1. Implementation of a logarithmic search and insertion symbol table (2 weeks)
2. Implementation of a lexical analyzer (2 weeks)
3. Implementation of a basic parser (2 weeks)
4. Implementation of a type checking system (2 weeks)
5. Implementation of an intermediate code generator (2 weeks)

Learning Outcomes:
• Knowledge and understanding
  - Understand the structure of compilers
  - Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation
  - Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines

• Cognitive skills (thinking and analysis).
  - Design and implement a compiler using a software engineering approach

• Communication skills (personal and academic).

• Practical and subject specific skills (Transferable Skills).
  - Use generators (e.g. Lex and Yacc)

Assessment Instruments

<table>
<thead>
<tr>
<th>Allocation of Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment Instruments</td>
</tr>
<tr>
<td>First examination</td>
</tr>
<tr>
<td>Second examination</td>
</tr>
<tr>
<td>Final Exam (written unseen exam)</td>
</tr>
<tr>
<td>Final project presentation</td>
</tr>
<tr>
<td>Reports, research projects, Quizzes, Home works, Projects</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>
* Make-up exams will be offered for valid reasons only with consent of the Dean. Make-up exams may be different from regular exams in content and format.

**Practical Submissions**
The assignments that have work to be assessed will be given to the students in separate documents including the due date and appropriate reading material.

**Documentation and academic honesty**
Submit your homework covered with a sheet containing your name, number, course title and number, and type and number of the homework (e.g. tutorial, assignment, and project).

Any completed homework must be handed in to my office (room ---) by 15:00 on the due date. After the deadline “zero” will be awarded. You must keep a duplicate copy of your work because it may be needed while the original is being marked.

You should hand in with your assignments:
1. A printed listing of your test programs.
2. A brief report to explain your findings.
3. Your solution of questions.

**Presentation**
On date due of your project, you will give a presentation of 15 minutes to the entire class describing your project and the progress you have made. You must practice in advance and make sure you take this amount of time (no more or no less). There will be an additional 5 minutes for questions by the class afterwards. Your presentation should include the following information (about 1-2 slides each):
   - Overview of project idea (motivation for it & overview of solution)
   - Tasks (example tasks that the system supports)
   - Demonstration (spend about 7-9 of your 15 minutes on this)

**Protection by Copyright**
1. Coursework, laboratory exercises, reports, and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.
2. Use of quotations or data from the work of others is entirely acceptable, and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.
3. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.

**Avoiding plagiarism.**
1. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
2. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and
changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.

3. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.

4. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

Course/Module Academic Calendar

<table>
<thead>
<tr>
<th>Week</th>
<th>Basic and support material to be covered</th>
<th>Homework/reports and their due dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Introduction to Compilers: The role of language translation in the programming process; Comparison of interpreters and compilers, language translation phases, machine-dependent and machine-independent aspects of translation, language translation as a software engineering activity</td>
<td>Tutorial 1</td>
</tr>
<tr>
<td>(2)</td>
<td>Lexical Analysis: Application of regular expressions in lexical scanners,</td>
<td>Tutorial 2</td>
</tr>
<tr>
<td>(3)</td>
<td>Lexical Analysis: hand coded scanner vs. automatically generated scanners</td>
<td>Tutorial 3, Lab project 1</td>
</tr>
<tr>
<td>(4)</td>
<td>Lexical Analysis: formal definition of tokens, implementation of finite state automata.</td>
<td>Tutorial 4</td>
</tr>
<tr>
<td>(5)</td>
<td>Syntax Analysis: Revision of formal definition of grammars, BNF and EBNF; bottom-up vs. top-down parsing.</td>
<td>Tutorial 5, Lab project 2</td>
</tr>
<tr>
<td>(6)</td>
<td>Syntax Analysis: tabular vs. recursive-descent parsers, error handling.</td>
<td>Tutorial 6</td>
</tr>
<tr>
<td>(7)</td>
<td>Parsers Implementation: automatic generation of tabular parsers, symbol table management, the use of tools in support of the translation process</td>
<td>Tutorial 7, Lab project 3</td>
</tr>
<tr>
<td>(8)</td>
<td>Semantic Analysis: Data type as set of values with set of operations, data types, type-checking models, semantic models of user-defined types, parametric polymorphism, subtype polymorphism, type-checking algorithms.</td>
<td>Tutorial 8, Lab project 4</td>
</tr>
<tr>
<td>(9)</td>
<td>Intermediate Representation, code generation: Intermediate and object code, intermediate representations, implementation of code generators</td>
<td>Tutorial 9</td>
</tr>
<tr>
<td>(10)</td>
<td>Code generation: code generation by tree</td>
<td>Tutorial 10, Lab project 5</td>
</tr>
<tr>
<td>Table Entries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(12) Error Detection and Recovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(13) Error Repair, Compiler Implementation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(14) Compiler design options and examples: C Compilers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(15) Specimen examination (Optional)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(16) Final Examination</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Expected workload:**
On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

**Attendance policy:**
Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

**Module References**
*Students will be expected to give the same attention to these references as given to the Module*