The course aims at teaching students about compiler development methodology and its associated technology to modern applications. The course contents included formal aspects, lexical analysis, syntax analysis, syntax-directed translation, run-time environments, intermediate code generation, code generation and code optimization.

本科旨在教授学生有关编译程序开发方法及其于现代应用之相关技术。本科内容包括形式方面、词法分析、语法分析、语法导向之翻译、运行时环境、中间代码之生成、代码生成及编码优化。

**Grade Descriptor:**

**A**

EXCELLENT – exceptionally good performance and far exceeding expectation in all or most of the course learning outcomes; demonstration of superior understanding of the subject matter, the ability to analyze problems and apply extensive knowledge, and skillful use of concepts and materials to derive proper solutions.

有關等級說明的資料，請參閱英文版本。

**B**

GOOD – good performance in all course learning outcomes and exceeding expectation in some of them; demonstration of good understanding of the subject matter and the ability to use proper concepts and materials to solve most of the problems encountered.

有關等級說明的資料，請參閱英文版本。

**C**

FAIR – adequate performance and meeting expectation in all course learning outcomes; demonstration of adequate understanding of the subject matter and the ability to solve simple problems.

有關等級說明的資料，請參閱英文版本。

**D**
MARGINAL – performance barely meets the expectation in the essential course learning outcomes; demonstration of partial understanding of the subject matter and the ability to solve simple problems.

有关等級說明的資料，請參閱英文版本。

F

FAILURE – performance does not meet the expectation in the essential course learning outcomes; demonstration of serious deficiencies and the need to retake the course.

有關等級說明的資料，請參閱英文版本。

Equivalent Offering:
Units: 3 (Min) / 3 (Max) / 3 (Acad Progress)
Grading Basis: Graded
Repeat for Credit: N
Multiple Enroll: N
Course Attributes:

Topics:

COURSE OUTCOMES

Learning Outcomes:
At the end of the course of studies, students will have acquired:
1. Fundamental concepts of the compiler, in particular, lexical analysis, syntax analysis, and code generation and optimization;
2. Deep understanding of how programming languages are executed in machines/computers;
3. Implementation experience in developing a real compiler for a certain programming language.

Course Syllabus:
Week 1: Course Overview and Project Introduction
Week 2: Formal Aspects in Compiler
Week 3: Lexical Analysis
Week 4: Syntax Analysis  
Week 5: Syntax-Directed Translation  
Week 6: Top-Down Parsing  
Week 7: Bottom-Up Parsing  
Week 8: Semantic Analysis  
Week 9: Run-time Environments  
Week 10: Intermediate Code Generation  
Week 11: Code Generation  
Week 12: Code Optimization  
Week 13: Course Summary and Project Presentation

Assessment Type:  
- Examination : 40%  
- Homework or assignment : 20%  
- Project : 40%

Feedback for Evaluation:  
1. Results of assignments and examination;  
2. Course evaluation and questionnaire;  
3. Reflection of teachers;  
4. Question-and-answer sessions during class;  
5. Student consultation during office hours or online; 

Required Readings:  
- 

Recommended Readings:  

OFFERINGS
1. CSCI3120
   Acad Organization=CSD; Acad Career=UG

COMPONENTS
LEC : Size=30; Final Exam=Y; Contact=3
TUT : Size=30; Final Exam=N; Contact=1

ENROLMENT REQUIREMENTS
1. CSCI3120
   Enrollment Requirement Group:
      Prerequisite: CSCI3130
   New Enrollment Requirement(s):
      Pre-requisite = CSCI3130

Additional Information

<table>
<thead>
<tr>
<th>eLearning hrs for blended cls</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTL-Onsite face-to-face hrs</td>
<td>0</td>
</tr>
<tr>
<td>VTL-Online synch. hrs</td>
<td>0</td>
</tr>
<tr>
<td>VTL-Online asynch. hrs</td>
<td>0</td>
</tr>
<tr>
<td>No. of micro-modules</td>
<td>0</td>
</tr>
<tr>
<td>Research components (UG)</td>
<td>0%</td>
</tr>
</tbody>
</table>