

1	Course Code:	COMP 4121
2	Course Title:	Modern Compiler Construction
3	Abbreviated Course Title:	modcc
4	Course Credits:	3
5	Class Quota:	60
6	Duration (1/2 sem or period in Summer):	1 semester (Spring 22-23)
7	Targeted Student Group:	CSE
8	Pre-requisite:	COMP 3021 OR COMP 3031
9	Exclusion:	N/A
10	Instructor's approval is required? (Y/N)	N
11	Instructor:	Lionel Parreaux
12	Area:	COMP
13	Course Description:	<p>Compiler implementation techniques are relevant to a wide array of software engineering areas. From parsing ad-hoc configuration file formats to validating complex specification languages to generating efficient code solving data-intensive problems at scale, many important problems of today and tomorrow require knowledge of basic compiler technology. The course exposes students to the essentials of modern compiler construction, including parsing, semantic analysis, program transformation, and code generation. Students will learn to design and implement their own programming language and extend it with a feature of their choice in a small team project. The course focuses on achieving these goals through effective high-level programming techniques, whose mastery will also make students better programmers in general.</p>
14	Tentative course structure: (optional)	<ol style="list-style-type: none"> <li>1. Overview, source languages and run-time models</li> <li>2. Review of formal languages</li> <li>3. Lexical analysis</li> <li>4. Syntactic analysis (parsing)</li> <li>5. Name analysis</li> <li>6. Type checking</li> <li>7. Type inference</li> <li>8. Code generation 1</li> <li>9. Code generation 2</li> <li>10. Optimization</li> </ol>

15	Intended learning outcomes (ILOs) of the course:	Designing and implementing computer languages, in particular simple programming languages; constructing and extending compilers; implementing language specifications; coordinating development with a project partner; delivering working programs.
16	Rationale for introducing the course:	CS department needs a compilers course
17	Planned teaching arrangement: (Lecture/Lab/Tutorial and the weekly offering pattern such as D2H1.5 for lecture and D1H1 for Lab, D/no. of day; H/no. of hour per week)	Lecture D1H2 Tutorial D1H1.5
18	Textbook / Reference books: (optional)	Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: Compilers: Principles, Techniques, and Tools (2nd Edition, 2006) Torben Mogensen, Basics of Compiler Design, (2010 edition, <a href="http://hjemmesider.diku.dk/~torbenm/Basics/">http://hjemmesider.diku.dk/~torbenm/Basics/</a> )
19	Grading Scheme	Projects 60% (five minor projects worth 5% each, and one major project worth 35%) Exam 40% Letter grade