1	Course Code:	COMP4901U
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	1 semester (Spring 22-23)
	period in Summer):	
7	Targeted Student	CSE
	Group:	
8	Pre-requisite:	COMP2012 or 2012H
9	Exclusion:	N/A
10	Instructor's approval is required? (Y/N)	Ν
11	Instructor:	Lonel Parreaux
12	Area:	СОМР
13	Course Description:	Compiler implementation encompasses a wide range of techniques which are relevant to many software engineering area. From parsing ad-hoc configuration file formats to validating complex specification languages to generating efficient code solving data-intensive problems at scale, may important problems of today and tomorrow fall within the general framework of compiler construction. In this context, the usefulness of compiler theory and practice goes much beyond understanding the technology underling your grandpa's old C compilers – our software-driven world is full of domain-specific languages that need correct and efficient implementations, and emergent areas like machine learning are more than ever in need of innovations from the field of compilation and optimization. In this course, you will learn the essentials of modern compiler construction, including parsing, analyzing, transforming, and code-generating programs. You will learn to design and implement your own programming language and extend it with features of your choice in a small team project.
	Tentative course structure: (optional)	 Overview, source languages and run-time models Review of formal languages Lexical analysis Syntactic analysis (parsing) Name analysis Type checking Type inference Code generation 1 Code generation 2 Optimization

COMP4901U Page 1 of 2 Spring 22-23

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	CS department needs a compilers course
	introducing the course:	
17	Planned teaching	Lecture D1H2
	arrangement:	Lab D1H2
	(Lecture/Lab/Tutorial	Tutorial D1H2
	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)	
18	Textbook / Reference	Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman:
	books:	Compilers: Principles, Techniques, and Tools (2nd Edition,
	(optional)	2006)
		Torben Mogensen, Basics of Compiler Design, (2010 edition,
		http://hjemmesider.diku.dk/~torbenm/Basics/)
19	Grading Scheme	Projects 60% (five minor projects worth 5% each, and one
		major project worth 35%)
		Exam 40%
		Letter grade