

SU DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
SYLLABUS (Tentative)
COSC 432 *Compiler Construction*

Description: Introduction to the principles, techniques and tools of modern compiler construction. Topics include lexical analysis, parsing, and semantic analysis, translation, code generation, and run time organization. Other topics to be discussed are abstract syntax, type checking, and register allocation. Students will design and implement a working compiler in this course. Three hours lecture per week.

Prerequisites: Advanced Data Structures (COSC320) with a grade C or better.

Co-requisites: Theory of Computing (COSC/MATH 362).

Reference: Writing Compilers and Interpreters - A Software Engineering Approach, 3rd edition by Ronald Mak (ISBN: 978-0-470-17707-5)

	Week
<i>Introduction</i>	2.0
Compiler and interpreter structure; Syntax and semantics; Lexical, syntax, and semantic analyses. Conceptual design (language-independent framework components: front end, intermediate tier and back end).	
<i>Scanning and Symbol Table</i>	2.0
Introduction to syntax diagrams: BNF (Backus-Naur Form); Deterministic Finite Automata (DFA), Tokenizer; Symbol table conceptual design and implementation;	
<i>Parsing Expressions and Statements</i>	3.0
Introduction to context free grammars, top-down and bottom-up parsing; Parsing expressions (arithmetic and logic) and statements (assignment, control and compound); Intermediate code design and implementation; Error handling.	
<i>Parsing Declarations, Type Checking, Programs, Procedures and Functions</i>	3.5
Parsing variable declarations; Data type specification, scope and symbol tables; Type checking expressions and statements. Program, procedure and function declaration and calls; Nested scopes, formal and actual parameters.	
<i>Jasmin Assembly Language and Code Generation for JVM</i>	1.5
Organization of Java Virtual Machine (JVM); Introduction to Jasmin instructions.	
<i>Code Generator</i>	2.0
Compiling statements (assignment, loop, control), procedures, functions and programs, procedure and function calls; Instruction selection; Register allocation; Runtime memory management (heap, stack, garbage collection); Code optimization (optional).	
<i>Total</i>	14.0

EVALUATION

Tests & Final Exams: 45 %

Project & Presentation: 55%