



**JOMO KENYATTA UNIVERSITY
OF
AGRICULTURE AND TECHNOLOGY**

BCT 2311 Compiler Construction ~ Course Outline

Course Goals

The students should be able to;

- Understand the fundamental issues in modern compiler design and implementation.
- Become familiar with concepts, methods, and tools in compiler front-end design.
- Become familiar with methods, techniques in compiler intermediate representation and code generation.
- Become familiar with basics of compiler analysis and optimization.

Course Description

Formal treatment of programming language translation and compiler design concepts; compilers and interpreters. Language theory. Parsing context-free languages, translation specifications and machine-independent code optimization; main phases of compilation. Lexical analysis. Syntax analysis. Semantic analysis. Code generation. Symbol table design. Program compilation. Loading and execution. Compilation techniques. Optimisation. Design of a simple complete compiler.

Course Plan

WEEK	DETAILS	HOURS
Week 1	Introduction: Introduction to compiler construction, Phases of compilers, Challenges in Compiler Design.	5
Week 2	Lexical Analysis: Role of lexical analyzer, specification of Tokens, Regular expressions, Lexical Analysis tools	5
Week 3	Lexical Analysis conti: Token recognition: Non-deterministic Finite Automata (NFA), Deterministic Finite Automata (DFA)	5
Week 4	Parsing And Parser Design: Role of parser, Grammar, Context- Free Grammar.	5
Week 5	Parsing And Parser Design conti: Top-Down: Recursive descent parsing, LL(1) parsing	5
Week 6	Parsing And Parser Design Conti: Bottom-Up parsing: Shift-reduce parsing, LR parsing, Types of LR parsing.	5
Week 7	Parsing And Parser Design Conti: LR(1) parsing, LR(1) parsing methods: SLR, Canonical LR, LALR	
Week 8	Semantic Analysis: Semantics, Attribute Grammar, Synthesized Attributes, Inherited Attributes	5
Week 9	Semantic Analysis cont: Type Checking: Static vs Dynamic checking, Type expressions	5
Week 10	Intermediate Code Generation: Intermediate Languages, Representation techniques, Syntax trees and P-code	5
Week 11	Intermediate Code Generation conti: Three address Code, One address code, Three address Code implementation, and Generation.	5
Week 12	Target Code Generation: Basic block, code generation for Tree, Register Allocation.	5
Week 13	Code Optimization: Classification of Optimization, Optimization Techniques, Optimization Transformations.	5
Week 14	Code Optimization conti: Local Optimization, Global Optimization, Data Flow Analysis.	5
Week 15 & 16	Examinations	

Reference Materials:

- Compilers: Principles, Techniques and Tools by Aho, Lam, Sethi, and Ullman; Addison-Wesley Pub Co, ISBN: 0201100886
- Compiler Construction: Principles and practices by Kenneth C. Loudon
- Basics of Compiler Design, Torben Mogensen
- Engineering a Compiler, 2nd edition, Keith D. Cooper and Linda Torczon
- Compiler Design by Santanu Chattopadhyay
- Internet Resources.

Delivery Mode: Lectures, Lab sessions, Assignments, and Reading

Method of Evaluation

- | | |
|---------------------------------|-----|
| • Written End of semester Exams | 70% |
| • Two Cats | 15% |
| • Assignment and Practical | 15% |

Lecturer: J. Wainaina