Master Course Syllabus
School of Engineering and Computer Science
Washington State University Vancouver
CS 452
Compiler Design
3 Semester Hours
(3 lecture hours)

Catalog Description

Design of lexical analyzers, syntactic analyzers, intermediate code generators, code optimizers and object code generators.

Prerequisite Courses

- CS 317 with a C or better
- CS 355 with a C or better

Prerequisite Topics

- Regular Expressions, Finite Automata, and Context Free Grammars (CFG)
- Basic programming language design concepts (e.g., type checking, scope, etc.)
- Proficiency in an imperative programming language and with object-oriented concepts

Measured Outcomes

Students taking this course will:

1. Modify CFG such that it is appropriate for LL and LR parsing schemes (contributes to performance criterion 6-a)
2. Construct a parser for a pedagogical programming language and generate code for a target execution model (contributes to performance criterion 6-d)
3. Use scanner and parser generator tools, e.g., FLEX, BISON (contributes to performance criterion 2-b)

Required Textbooks

- Engineering a Compiler by Keith D. Cooper and Linda Torczon, 3rd edition 2023, Morgan Kaufmann

Reference Material

- Compilers: Principles, Techniques, and Tools by Aho, Lam, Sethi and Ullman, Addison Wesley

Major Topics Covered in the Course

1. Lexical analysis, Syntactic analysis, Semantic Analysis
2. Symbol Table
3. Intermediate Representation
4. Optimization
5. Code Generation

Projects

Programming Project Area	Weeks
C to HTML translator focusing on Lexical Analysis	2
Top-Down LL Parsing to implement an Interpreter	3
Bottom-Up LR Parsing and Simple Code Generation to implement a simple Compiler	3

Design, Implementation and Analysis

Students will learn to identify all essential steps for translating source code into assembly or other low-level language. Programming projects and quizzes will evaluate students on the major course outcomes.

Course Coordinator:	Farhana Kabir
Last Updated:	August, 2024
Syllabus Version Number:	1.0