CSE 5343 (Approved): Compiler Design and Implementation

Course Description

Lexical and syntax analyses using compiler generation tools; type checking; intermediate code; control-flow analysis; dataflow analysis; code optimizations; code generation; compiler project.

Prior Course Number: 756
Transcript Abbreviation: Cmplr Des & Impl
Grading Plan: Letter Grade
Course Deliveries: Classroom
Course Levels: Undergrad, Graduate
Student Ranks: Senior, Masters, Doctoral
Course Offerings: Spring
Flex Scheduled Course: Never
Course Frequency: Every Year
Course Length: 14 Week
Credits: 3.0
Repeatable: No
Time Distribution: 3.0 hr Lec
Expected out-of-class hours per week: 6.0
Graded Component: Lecture
Credit by Examination: No
Admission Condition: No
Off Campus: Never
Campus Locations: Columbus
Prerequisites and Co-requisites: (CSE 3901 or CSE 3902 or CSE 560) and (CSE 3341 or CSE 655)
Exclusions: Not open to students with credit for CSE 756
Cross-Listings:

The course is required for this unit's degrees, majors, and/or minors: No
The course is a GEC: No
The course is an elective (for this or other units) or is a service course for other units: Yes

Subject/CIP Code: 14.0901
Subsidy Level: Doctoral Course

Programs

Abbreviation	Description
BS CSE	BS Computer Science and Engineering
MS CSE	MS Computer Science and Engineering
PhD CSE	PhD Computer Science and Engineering

Course Goals

Master using tools for generation of lexical analyzers and parsers;
Master generating intermediate code;
Be competent with control-flow and dataflow analysis;
Be competent with simple code optimizations;
Be familiar with techniques for top-down and bottom-up parsing;
Be familiar with type checking;

Be familiar with generation of machine code;
Be familiar with optimizations for parallelism and locality;
Be exposed to techniques for lexical analysis;
Be exposed to register allocation;
Be exposed to instruction scheduling.

Course Topics

Topic	Lec	Rec	Lab	Cli	IS	Sem	FE	Wor
Compiler structure	2.0							
Lexical analysis	3.0							
Parsing	3.0							
Type checking	3.0							
Intermediate code	6.0							
Control-flow analysis	3.0							
Dataflow analysis	3.0							
Code optimizations	6.0							
Generation of machine code	6.0							
Parallelism and locality	3.0							
Instruction scheduling	2.0							
Register allocation	2.0							

Representative Assignments

Implement a lexical analyzer;
Implement a parser;
Implement type checking;
Implement generation of intermediate code;
Implement control-flow analysis;
Implement dataflow analysis;
Implement simple code optimizations.

Grades

Aspect	Percent
Lab assignments	75%
Final exam	25%

Representative Textbooks and Other Course Materials

Title	Author
Compilers: Principles, Techniques, and Tools, Second Edition	Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman

ABET-EAC Criterion 3 Outcomes

Course Contribution		College Outcome
***	a	An ability to apply knowledge of mathematics, science, and engineering.
***	b	An ability to design and conduct experiments, as well as to analyze and interpret data.
***	c	An ability to design a system, component, or process to meet desired needs.
*	d	An ability to function on multi-disciplinary teams.
***	e	An ability to identify, formulate, and solve engineering problems.
*	f	An understanding of professional and ethical responsibility.
*	g	An ability to communicate effectively.
	h	The broad education necessary to understand the impact of engineering solutions in a global and societal context.
**	i	A recognition of the need for, and an ability to engage in life-long learning.
	j	A knowledge of contemporary issues.
***	k	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

BS CSE Program Outcomes

Course Contribution		Program Outcome
***	a	an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
***	b	an ability to design and conduct experiments, as well as to analyze and interpret data;
***	c	an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
*	d	an ability to function on multi-disciplinary teams;
***	e	an ability to identify, formulate, and solve engineering problems;
*	f	an understanding of professional, ethical, legal, security and social issues and responsibilities;
*	g	an ability to communicate effectively with a range of audiences;
	h	an ability to analyze the local and global impact of computing on individuals, organizations, and society;
**	i	a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
	j	a knowledge of contemporary issues;
***	k	an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
***	l	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
***	m	an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
***	n	an ability to apply design and development principles in the construction of software systems of varying complexity.

Prepared by: Neelam Soundarajan