University at Buffalo
The State University of New York

CSE443 -Compilers
Spring 2025

COURSE INFORMATION

Lecture times – MWF 10:00 – 10:50
Location – Norton 210

Number of credits (include type of credits if applicable): 4

Instructor(s) names and contact information:
Dr. Carl Alphonce
e-mail: alphonce@buffalo.edu

On-line appointment: https://calendly.com/alphonce
Office hours: Tuesdays 9:00 AM – 11:00 AM

COURSE DESCRIPTION

Considers problems encountered in the design and implementation of a translator for high-level programming languages: lexical analysis, context-free grammars, parsing, storage allocation, code generation and optimization, and error recovery. Uses compiler construction tools for the programming projects.

Course Pre-requisite: CSE220 Systems Programming
Course Co-requisite: CSE341 Computer Organization

STUDENT LEARNING OUTCOMES

RELEVANT PROGRAM OUTCOMES

(ABET CAC 2) An ability to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
(ABET CAC 5) An ability to function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
(ABET CAC 6) An ability to apply computer science theory and software development fundamentals to produce computing-based solutions.

ABET CAC Student Outcome support (CS):

STUDENT OUTCOME	1	2	3	4	5	6
SUPPORT LEVEL	0	3	0	0	3	3

(ABET EAC 1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
(ABET EAC 5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

ABET EAC Student Outcome support (CEN):

STUDENT OUTCOME	1	2	3	4	5	6	7
SUPPORT LEVEL	3	0	0	0	3	0	0

Course Learning Outcome	Program Outcomes & Competencies		Instructional Method(s)	Assessment Method(s)
	CEN	CS		
Identify and describe the function of the major phases of a compiler.	(1)	(2)(6)	Lecture-based instruction Hands-on activities in lecture	EX
Define formally the grammars used in the front end of a compiler, their application in the front end, and techniques for parsing such grammars.				
Evaluate (compare and contrast) different intermediate representations.				
Explain the compiler's role in creating and managing run-time environments.				
Explain and evaluate (compare and contrast) different approaches to code generation.				
Explain the applicability and operation of code optimizations.				
Build both the front and back ends of a compiler (application of above theory)	(1)(5)	(2)(5)(6)	Lecture-based instruction Team project w/team-faculty meetings	PRJ-FUN
Collaborate effectively as a member of a software development team	(5)	(5)	Lecture-based instruction Team project w/team-faculty meetings	PRJ-PRO

Course Requirements

- (PRJ) Project (with 4 sprints before a final submission and weekly team meetings with project managers). Assessment is based on several components:
    - PRJ-FUN: the functionality of the project
    - PRJ-PRO: student engagement in the team software development process, including peer evaluation
    - PRJ-PRE: student engagement and performance in an in-class oral presentation/demo
- (EX) Cumulative final exam (during the final exam period) – an individual essay-based final examination

Grading Policy

Component weighting

Weight	Assessment / Assignment
50%	PRJ-FUN: project (functionality)
20%	PRJ-PRO: teamwork process engagement
10%	PRJ-PRE: project demo/presentation
20%	EX: cumulative final exam

Course Grade

Grade	Quality Points	Percentage
A	4.0	93.0% -100.00%
A-	3.67	90.0% - 92.9%
B+	3.33	87.0% - 89.9%
B	3.00	83.0% - 86.9%
B-	2.67	80.0% - 82.9%
C+	2.33	77.0% - 79.9%
C	2.00	73.0% - 76.9%
C-	1.67	70.0% - 72.9%
D+	1.33	67.0% - 69.9%
D	1.00	60.0% - 66.9%
F	0	59.9 or below

Incompletes (I/IU): Unless superseded by changes in university policy, a grade of incomplete ("I") indicates that additional course work is required to fulfill the requirements of a given course. Students may only be given an "I" grade if they have a passing average in coursework that has been completed and have well-defined parameters to complete the course requirements that could result in a grade better than the default grade. An "I" grade may not be assigned to a student who did not attend the course.

ACADEMIC INTEGRITY

Academic integrity is a fundamental university value. Through the honest completion of academic work, students sustain the integrity of the university while facilitating the university's imperative for the transmission of knowledge and culture based upon the generation of new and innovative ideas. See the Academic Integrity Policies of the university and the CSE department for details:
catalog.buffalo.edu/policies/integrity.html
engineering.buffalo.edu/computer-science-engineering/information-for-students/policies/academic-integrity.html

ACCESSIBILITY RESOURCES

Accessibility Resources coordinates reasonable accommodations for equitable access to UB for students with disabilities. www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html

COUNSELING SERVICES

Counseling Services can help with emotional issues, stress, crisis management and much more to support mental wellness through a variety of services. www.buffalo.edu/studentlife/who-we-are/departments/counseling.html

COURSE ORGANIZATION / TENTATIVE SCHEDULE (SUBJECT TO CHANGE – A DAY-BY-DAY SCHEDULE APPEARS ON COURSE WEBSITE)

Week #	Topic	Required Readings / Assignments(s)
1-2	Overview / Teamwork / Lexical analysis	Chapters 1, 2 / Chapter 3
3-4	Syntactic analysis	Chapter 4
5-6	Syntax-directed translation	Chapter 5
7-8	Intermediate code generation	Chapter 6
9-10	Runtime environments	Chapter 7
11-12	Code generation	Chapter 8
13-14	Machine-independent optimizations	Chapter 9

COURSE MATERIALS

- *Compilers: principles, techniques, and tools* (2nd edition), Aho, Lam, Sethi, Ullman. Pearson Education, 2007.

VALUES STATEMENT

The Department of Computer Science and Engineering at the University at Buffalo is a community dedicated to supporting excellence in scholarship and professionalism in all areas of computing. As a community we are bound together by humanity, diversity, equity, inclusiveness, and integrity.

Humanity embodies the ideal that all people are worthy of respect and dignity.

Diversity celebrates that every lived experience informs and can give voice to new discoveries, the lifeblood of innovation.

Equity recognizes that opportunities must be accessible to all.

Inclusiveness ensures that all are welcome and know they are valued members of the CSE community.

Integrity is the obligation to earn and maintain the trust of others.

In concert, these ideals are the foundation for effecting positive change in the world and contribute to personal and professional growth and success.