CSE 420 Syllabus, Schedule, and Protocols

Fall 2024 Version

Contents

[Rules and Guidelines 1](#_Toc180173285)

[Use of the Reference Textbook 1](#_Toc180173286)

[Self-study Policy 1](#_Toc180173287)

[Medical Emergency Reporting Guidelines 2](#_Toc180173288)

[Theory Assignment Late Submission Policy 2](#_Toc180173289)

[Theory Lecture Schedule 2](#_Toc180173290)

[Lecture Schedules for the Midterm Exam 2](#_Toc180173291)

[Lecture Schedule for the Final Exam 5](#_Toc180173292)

**Reference Textbook:** Compilers Principles, Techniques, and Tools (Second Edition) by Aho, Lam, Sethi, and Ullman

# Rules and Guidelines

## Use of the Reference Textbook

Note that the reference textbook by Aho and Ullman is considered the best book to teach compiler design all over the world and is currently in use in numerous universities. In prominent universities, sometimes there are more than one reference books, however, this book is always present and bulk of the materials are always picked from it. The reason is that the book is among the oldest and most detailed compiler related book one can found. Students will definitely find good explanation and detail of any materials discussed in the class in this book. Consequently, students must avoid any assumption that they do not have adequate materials to study the concepts and approaches discussed in the classroom or to get clarification about implementation hints regarding lab assignments. **Everything is available in the book and studying the book is essential for doing well in the course**.

However, given the book is very detailed and rigorous, a mapping between the lecture content and sections/subsections of the book is provided for each lecture so that students do not have to search relevant items. In addition, some simplifying alternative reading materials are given for a few lectures for which the book content is scattered in too many places.

Students are recommended to come to the consultation hours of the theory/lab instructors if they find difficulty understanding any specific content of the reference textbook. However, they must never assume that some videos or slides can replace the need of the reference textbook.

## Self-study Policy

The compiler design course is designed to give a comprehensive experience of understanding the theory and practice of compiler constructions through the theory classes and lab assignments. To keep an alignment between the lab and theory in a short trimester system under such objective, details of some concepts are skipped in theory classes. However, to have a holistic knowledge of a programming language compiler construction, students should know those details. The policy we adopt to instill the knowledge of detail is by assigning some **self-study** theory assignments. Understand that, these are self-study topics. That means, students must study some sections/subsections of the book **on their own** and solve the assignment problems. Students are allowed to do group studies, and are welcome to take help about theory assignments during the consultation hours. However, they must never plagiarize their assignment submissions. Any plagiarized submission will get a straight zero.

## Medical Emergency Reporting Guidelines

A student can miss a theory quiz or assignment deadline due to medical emergencies. Then the section instructor can decide to accept a late submission of assignment or arrange a makeup quiz for the student. However, this is solely under the discretion of the section instructor whether to provide a student with such options. Students must understand that sickness and travails are part of life and one needs to make progress in study regardless of their presence. In addition, note that medical report for sickness will be accepted for the student him/herself only. Sickness of family members or relatives are not acceptable excuse for missing any quiz or assignment deadline. Furthermore, a student must report about his/her sickness as soon as possible after recovering from the illness. A medical report shared much late will be ignored.

## Theory Assignment Late Submission Policy

Note that since lab assignments are done in pair, by-weekly, and incremental; late submission of lab assignments are not accepted in any circumstances. Regarding the late submission of theory assignments, note that if the assignment is to be submitted using some online platform (Google forms, Google classroom, or Turnitin) students are responsible for submitting the assignments early enough so that any internet connectivity issue or server system glitch does not affect them. That is, they should not wait till the last 10 minutes of the deadline and run the risk of failing to submit their work even after completing it. Students must understand that instructors do not have any control over Google’s or Turnitin’s server. So that they cannot fix any system issue for them. Furthermore, each theory assignment is given two weeks to complete even though none requires more than 4 hours of work only to ensure that students have enough free time to work on them and to submit.

# Theory Lecture Schedule

## Lecture Schedules for the Midterm Exam

**Lecture 1: Introduction to Compiler**

The Structure and stages of the compiler

The analysis-synthesis model of compilation

Difference between a compiler and an interpreter

Reading References:

1. <https://betterprogramming.pub/compiler-vs-interpreter-d0a12ca1c1b6?gi=4ace49f83583>
2. Lesson 1 PDF file shared here
3. Chapter 1: Sections 1.2 to 1.5 (Inclusive) of the reference textbook

**Lecture 2: Introduction to Lexical Analysis**

The role of a Lexical Analyzer

Tokens, Patterns, Token Attributes

Regular Definitions

The Structure of a Generated Lexical-Analyzer

**Take-home assignment-1:** an assessment of students’ knowledge of regular expression, RE to NFA, then NFA to DFA from Automata and Complexity course

Reading References:

1. <https://www3.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html>  
   <https://cs.lmu.edu/~ray/notes/regex/>
2. Lesson 2 PDF file shared here
3. Chapter 3 Sections 3.1 and 3.3 of the reference textbook

**Self-Study Assignment 1:** students have to study the direct method of converting regular expression to DFA from Subsections 3.9.1 to 3.9.5 of Chapter 3 of the reference textbook and show the process of DFA construction for a reference regular expression. (Time 2 weeks)

**Lecture 3: Introduction to Syntax Analysis**

The role of a parser

Context Free Grammars

Parse Trees and Derivations

Grammar Ambiguity and Mitigation Techniques

Verifying the Language Generated by a grammar

Reading References:

1. Chapter 4 Section 4.1 and subsection 4.2.1 to 4.2.5 of the reference textbook
2. Lesson 3 PDF file shared here

**Lecture 4: Introduction to Bottom-up Parsing**

Concept of Reductions

Stack simulation example of Bottom-up Parsing

Concept of Shift-reduce Parsing

Handles and Handle Pruning

Reading References:

1. Chapter 4 Section 4.5 of the reference textbook

**Lecture 5: Introduction to Simple LR Parsing**

LR(0) Items

Closure of a LR(0) Item Set

LR(0) Automaton Construction

Reading References:

1. Chapter 4 subsection 4.6.1 and 4.6.2 of the reference textbook

**Lecture 6: Construction of SLR Parsing Table**

FIRST and FOLLOW computation

Generation of SLR Parsing Table from LR(0) automaton and FOLLOW set of non-terminals

Reading References:

1. Chapter 4 Subsection 4.4.2 and 4.6.4 of the reference textbook
2. Chapter 4 Subsection 4.6.3 (Only the structure of LR parsing table) of the reference textbook

**Lecture 7: SLR Parsing Algorithm**

Behavior of a SLR Parser

Example of SLR Parsing

Shift-reduce/Reduce-reduce Conflicts and Other Errors in Bottom-up Parsing

Reading References:

1. Chapter 4 Subsection 4.6.3 and 4.8.3 of the reference textbook

**Self-Study Assignment 2:** students have to study the canonical LR(1), that is, CLR(1), parser construction from Section 4.7.1 to 4.7.3 of the reference textbook and show the process of CLR(1) parser construction from a small reference grammar. (Time 2 weeks)

**Lecture 8: Symbol Table**

Importance of Symbol Tables

Scopes in Program and Nesting of Symbol Tables

Important attributes of variables and functions

Reading References:

1. Lesson 8 Slides shared here that are prepared by Professor Rich Maclin of the University of Minnesota Duluth
2. <https://pages.cs.wisc.edu/~fischer/cs536.s08/course.hold/html/NOTES/6.SYMBOL-TABLES.html>

**Lecture 9:** **Introduction to Syntax-directed Translation**

The logic of syntax directed translation

Inherited and Synthesized Attributes

Evaluation of syntax directed definitions

Reading References:

1. Lesson 9 PDF file shared here
2. From start of the Chapter to Example 5.2 in Page 307 of the reference textbook
3. From the start of Section 5.4 to Example 5.11 in Page 318 of the reference textbook

**Lecture 10: Construction of the Parse Tree using SDD Rules**

Importance of a Syntax Tree

SDT/SDT for Syntax Trees

Construction Example of a Syntax Tree

One class equivalent time is reserved for two quizzes before the Midterm exam. If 12 classes can be taken before the midterm, then the other class should be for reviewing so far materials.

## Lecture Schedule for the Final Exam

**Lecture 11: Dependency Graphs and SDD Types**

Dependency Graphs

S-attributed Definitions

L-attributed Definitions

Reading References:

1. Section 5.2 from Chapter 5 of the reference text book

**Lecture 12: Type Grammar and Attributes of a Type**

Type Expression Grammar

Attributes of an Array Type

Grammar for a sequence of variable declarations

Grammar and attributes of Record Types

Reading References:

1. Section 5.3.2 of Chapter 5 and Section 6.3 of Chapter 6 of the reference textbook

**Lecture 13: Type Information Processing During Bottom-up Parsing**

SDT for Array Type Calculation

SDT for variable widths and offsets calculation in a sequence of variable declarations

Reading References:

1. Same as Lecture 13

**Lecture 14: Introduction to Intermediate Code Generation**

Importance of three address codes as Intermediate Representation

Features of three address codes

Quadruples

Triples

Reading References:

1. Chapter 6 from Introduction up to Section 6.2 (Inclusive) of the reference textbook

**Lecture 15: Array Access Logic and SDT for Array Accesses**

Addressing an array element

Translation of Array References

Reading References:

1. Chapter 6 Section 6.4 of the reference textbook

**Lecture 16-17: Handling Flow-of-Control Statements**

Grammar for branching and loops

SDT for translating if-else and while loops

Reading References:

1. Chapter 6 Section 6.6 of the reference textbook

**Self-Study Assignment 3:** students have to study the intermediate code generation for procedure from Section 6.9 of the reference textbook and generate the three-address code for a code block containing loops and nested procedure calls. (Time 2 weeks)

**Lecture 18-19**: **Handling Object Oriented Language Features during Compilation**

Handling class object’s field access

Translation of class methods

The logic of dynamic dispatch

Reading References:

1. Lesson 18 PDF file shared here

**Lecture 20: Introduction to Runtime Storage Organization**

Different parts of a process memory

Memory Layout

The content of stack and heap

Reading References:

1. Chapter 7 Section 7.1 of the reference text book

Two class equivalent time is reserved for the two quizzes before the Final exam and for reviewing materials discussed after the midterm.