# CSC 453, Compilers and Systems Software

## Fall 2025

**Meeting times: TuTh 12:30-1:45 PM**
**Classroom: Harvill Bldg, Room 302**

## Course Description

This course covers the design and implementation of translator-oriented systems software such as compilers and interpreters. Topics covered include lexical analysis, parsing, and syntax-directed code generation. ***This class has a significant programming component.***

The official UA Catalog description of this course is available here.

## Instructor and Contact Information

**Instructor:** Saumya Debray
Email: debray@arizona.edu
Office Hours: See class site in D2L

**Teaching Assistants:** Bennett Brixen and Phyllis Spence
Email**:** `csc453@list.arizona.edu` [reaches both TAs as well as the instructor]
Office Hours: See class site in D2L

## Course Format and Teaching Methods

Lecture only

## Course Objectives

- Lexical analysis;
- Context-free grammars and recursive-descent parsing;
- Symbol tables;
- Types and type checking;
- Intermediate representations of code: abstract syntax trees, three-address code;
- Code generation via syntax-directed translation;
- Advanced topics (as time permits): interpreters and JIT compilers; binary code files.

## Expected Learning Outcomes

Students who successfully complete this course should be able to understand and explain:

- the structure of compilers; phases of compilation, including lexical analysis (scanning), syntax analysis (parsing), semantic analysis, intermediate and final code generation;
- the use of regular expressions to specify lexical tokens and the use of finite state automata to recognize tokens;
- the use of context-free grammars to specify programming language syntax, and the use of parsers to recognize context-free languages;
- intermediate representations, including abstract syntax trees and three-address code;
- syntax-directed translation and code generation;
- interpreters and interpretation; Just-in-time (JIT) compilation.

As part of the course, students will implement a compiler for a significant subset of C.

## Transferable Career Skills

- **Critical thinking**. Students will use their understanding of theoretical and practical aspects of compiler design to make appropriate design and implementation decisions for data structures and algorithms and thereby construct a compiler for a significant subset of C.
- **Technology.** Students will understand and leverage technology to write, test, and debug a large body of code.
- **Professionalism**. Students will demonstrate effective work habits to complete a large and complex project consisting of multiple implementation milestones with tight deadlines.

## Makeup Policy for Students Who Register Late

Students who register after the first class meeting may not make up any missed work.

## Course Communications

Course communications will be made through:

- **D2L** [course materials and info]**:** https://d2l.arizona.edu/d2l/home/1630734
- **Gradescope** [assignments]**:** https://www.gradescope.com/courses/1065263
- **Discord** [questions and discussion]**:** https://discord.gg/ZuPwggbVPJ

If you are not already enrolled in the Gradescope and Discord sites for this class, please do so right away (see the announcement in D2L for instructions on how to self-enroll).

## Required Texts or Readings

This class has no required text. The primary reference material will be the instructor's lecture notes, which will be made available to students on the class website. Additionally, students may use the following book (available online for free) as an optional secondary reference:

> Torben Mogensen. *Basics of Compiler Design*. Available at http://www.diku.dk/~torbenm/Basics/index.html.

## Assignments and Examinations: Schedule/Due Dates

### *Programming Assignments*

The course has a programming project where students implement a complete compiler for a subset of the C programming language. This project is divided into a number of programming assignments, each of which builds on all of the previous assignments. Each assignment further consists of a sequence of milestones implementing different components of the compiler.

You will submit the code for each assignment at the submission portal for that assignment in GradeScope. You can submit your code as many times as you want prior to the submission deadline. Your submission will be auto-graded when it is uploaded and you will receive your score within a few minutes of submission (barring exceptional circumstances beyond the instructor's control, e.g., if the GradeScope website crashes).

**Collaboration:** All assignments are individual (i.e., there is no partnering).

**Timeliness:** Assignments are due at or before the submission deadline given on the assignment spec. Late submissions will not be accepted.

### *Programming assignments: what's allowed and what isn't*

It is permissible to discuss problems with others in broad terms, e.g., the structure or approach of a program. It is not permissible to discuss concrete details of solutions to a particular assignment before the due date/time for that assignment. In other words, you can talk to each other in English, but not in C/Unix.

The work you turn in for credit should be substantially your own. It is permissible to share test inputs with other students; collaboration beyond this on programming assignments is not permitted.

It is permissible to use modest amounts of "publicly visible" code—code that is available in books or magazines, or which has been distributed/discussed in class—in programming assignments, *as long as the authorship of such code is adequately and explicitly acknowledged*. It is not permissible to solicit code from others. It is also not permissible to use code written by CSc 453 students in previous terms. Please check with me or the TA ahead of time if you'd like to use someone else's code in order to make sure that the amount of code is indeed modest.

For the purposes of this course, cheating is considered to be any attempt to pass off someone else's work as your own. Cheating will not be tolerated: any student caught cheating or helping another student cheat in homeworks, exams, or programming assignments, will be given a failing grade in the course. I interpret the phrase "helping another student cheat" broadly: e.g., if another student gains access to your code because you forgot to logout, or were careless about listings that were dumped into the recycling bin, you have helped that student cheat. For the same reason, you should be very careful about posting your code to publicly visible media, e.g., Discord or Github.

## *Use of AI tools*

The use of generative AI tools, such as ChatGPT, Dall-e, Google Bard, etc., is NOT allowed for any purpose in this course and will be considered a violation of the Code of Academic Integrity, specifically the prohibition against submitting work that is not your own. This applies to all assessments in the course, including programming assignments, quizzes, exams, and homework problem sets. In particular, the following actions are prohibited:

- entering all or any part of an assignment statement or test questions as part of a prompt to a large language model AI tool;
- incorporating any part of an AI-written response in an assignment;
- using AI to summarize or contextualize reading assignments or source materials; and
- submitting your own work for this class to a large language model AI tool for iteration or improvement.

## *Exams*

- **Midterm exam**: Tue Oct 14, 12:30 PM - 1:45 PM (75 mins)
- **Final exam**: Wed Dec 17, 1:00 PM - 2:15 PM (75 mins)

Without prior arrangements, missed exams result in a grade of zero. If you will be absent on the date of an exam due to religious reasons or because of a pre-approved absence by the Dean of Students, please contact me ahead of time so that we can work out an alternative time for your exam.

## *Final Examination*

- Date and time: **Wed Dec 17**, 1:00 PM to 2:15 PM
- Final Exam Regulations and Final Exam Schedule: https://registrar.arizona.edu/finals

## *Assignment Schedule*

The schedule for these assignments is as follows (**Note**: this is subject to change with advance notice):

| Assg No. | Milestone | Topic | Start date | Due date |
|---|---|---|---|---|
| 1 (2 weeks) | 1 | Lexical analysis (full C-- language) | 09/04/2025 | 09/10/2025 |
| | 2 | Parsing for G0 subset of C-- | 09/11/2025 | 09/17/2025 |
| 1 day | | | | |
| 2 (3.2 weeks) | 1 | Parsing for G1 subset of C-- | 09/19/2025 | 09/25/2025 |
| | 2 | Parsing and Type checking for G2 | 09/26/2025 | 10/02/2025 |
| | 3 | AST construction G2 subset of C-- | 10/03/2025 | 10/09/2025 |

| | | | | |
|---|---|---|---|---|
| | 6 days | | | |
| 3 (2.2 weeks) | 1 | Final code generation for a subset of G2 | 10/16/2025 | 10/27/2025 |
| | 2 | Final code generation for G2 | 10/27/2025 | 11/10/2025 |
| | 1 day | | | |
| 4 (3.5 weeks) | 1 | Parser for full C-- language | 11/12/2025 | 11/19/2025 |
| | 2 | Type checking + AST generation for C-- | 11/19/2025 | 11/26/2025 |
| | 3 | Final code generation for full C-- | 12/01/2025 | 12/08/2025 |

## Grading Scale and Policies

Grades will be computed based on the following weights for the various components of the class:

| | |
|---|---|
| Homework assignments | 10% |
| Midterm exam | 20% |
| Final exam | 20% |
| Programming assignments<br>(Note: different assignments have different degrees of difficulty and therefore have different weights. See D2L for details.) | 50% |

Your grade will be determined by the overall weighted average of your scores, computed using the weights given above, based on the following mapping:

| **Weighted average** | **Grade** |
|---|---|
| ≥ 90 | A |
| ≥ 80 but below 90 | B |
| ≥ 70 but below 80 | C |
| ≥ 60 but below 70 | D |
| < 60 | E |

University policy regarding grades and grading systems is available at
https://catalog.arizona.edu/policy/courses-credit/grading/grading-system

## Grading procedure

The size of the programs involved in the project makes it impractical to manually examine your source code to determine its correctness. Instead, we will use the following procedure:

- You will develop and test your code using your own test cases. It is permissible for students to share test cases.
- I will make my test cases public after the submission deadline.
- Your code will be graded on my test cases using a grading script. You will be awarded a preliminary score based on the number of test cases failed.

- If you were penalized more than once for the same problem, you will have the option of bringing this to my attention over the two weeks following notification of your preliminary score. Specifically, you will need to provide me with the following items (you can do this either via email or in person):
    - a list of the specific problems in your code; and
    - for each problem, the test cases that failed as a result.
- Based on this, I may adjust your preliminary score where appropriate, based on my assessment of the seriousness of the problems. *However, any such adjustment will always be positive, i.e., you will not be penalized—but may be rewarded—for identifying and explaining the problems in your code.*

## Incomplete (I) or Withdrawal (W):

Requests for incomplete (I) or withdrawal (W) must be made in accordance with University policies, which are available at https://catalog.arizona.edu/policy/courses-credit/grading/grading-system.

## Dispute of Grade Policy:

All regrade requests for programming assignments must be made within two weeks of when the grade is returned. All regrade requests for midterm exams must be made within one week of when the graded exam is returned.

## Scheduled Topics and Activities

See lecture slides (posted to D2L) for reference

| Week no. | Week of | Lecture Topic | Assignments and Exams |
|---|---|---|---|
| 1 | 08/25 to 08/29 | Compilers: overview; Lexical analysis (scanning) | |
| 2 | 09/01 to 09/05 | Lexical analysis (scanning) | 09/04: Assg 1 MS 1 start |
| 3 | 09/08 to 09/12 | Context-free grammars; Recursive descent parsing | 09/10: Assg 1 MS 1 due<br>09/11: Assg 1 MS 2 start |
| 4 | 09/15 to 09/19 | Recursive descent parsing | 09/17: Assg 1 MS 2 due<br>09/19: Assg 2 MS 1 start |
| 5 | 09/22 to 09/26 | Semantic checking | 09/25: Assg 2 MS 1 due<br>09/26: Assg 2 MS 2 start |
| 6 | 09/29 to 10/03 | Abstract syntax trees (ASTs); Runtime environments | 10/02: Assg 2 MS 2 due<br>10/03: Assg 2 MS 3 start |
| 7 | 10/06 to 10/10 | ASTs and code generation | 10/09: Assg 2 MS 3 due |
| 8 | 10/13 to 10/17 | Code generation | **10/14 : Midterm exam**<br>10/16: Assg 3 MS 1 start |
| 9 | 10/20 to 10/24 | Theory: ambiguity, FIRST and FOLLOW sets, LL(1) grammars | |
| 10 | 10/27 to 10/31 | Recursive descent parsing: associativity and precedence | 10/27: Assg 3 MS 1 due<br>10/27: Assg 3 MS 2 start |
| 11 | 11/03 to 11/07 | Recursive descent parsing: associativity and precedence | |
| 12 | 11/10 to 11/14 | Short-circuit evaluation of Boolean expressions<br>(Note: No class on Tue 11/11: Veteran's Day) | 11/10: Assg 3 MS 2 due<br>11/12: Assg 4 MS 1 start |
| 13 | 11/17 to 11/21 | Code generation for expressions | 11/19: Assg 4 MS 1 due<br>11/19: Assg 4 MS 2 start |
| 14 | 11/24 to 11/26 | Interpreters | 11/26: Assg 4 MS 2 due |
| | 11/27 to 11/30 | Thanksgiving Break | |
| 15 | 12/01 to 12/05 | JIT compilers | 12/01: Assg 4 MS 3 start |

| | | | |
|---|---|---|---|
| 16 | 12/08 to 12/10 | review | 12/08: Assg 4 MS 3 due<br>12/10: last day of classes |
| | 12/17 | | **12/17: Final exam** |

## Classroom Behavior Policy

To foster a positive learning environment, students and instructors have a shared responsibility. We want a safe, welcoming, and inclusive environment where all of us feel comfortable with each other and where we can challenge ourselves to succeed. To that end, our focus is on the tasks at hand and not on extraneous activities (e.g., texting, chatting, reading a newspaper, making phone calls, web surfing, etc.).

Students are asked to refrain from disruptive conversations with people sitting around them during lecture. Students observed engaging in disruptive activity will be asked to cease this behavior. Those who continue to disrupt the class will be asked to leave lecture or discussion and may be reported to the Dean of Students.

## Safety on Campus and in the Classroom

For a list of emergency procedures for all types of incidents, please visit the website of the Critical Incident Response Team (CIRT): https://cirt.arizona.edu/case-emergency/overview

Also watch the video available at
https://arizona.sabacloud.com/Saba/Web_spf/NA7P1PRD161/app/me/ledetail;spf-url=common%2Flearningeventdetail%2Fcrtfy000000000003841

## University-wide Policies link

Links to the following UA policies are provided here, http://catalog.arizona.edu/syllabus-policies:

- Absence and Class Participation Policies
- Threatening Behavior Policy
- Accessibility and Accommodations Policy
- Code of Academic Integrity
- Nondiscrimination and Anti-Harassment Policy

## Department-wide Syllabus Policies and Resources link

Links to the following departmental syllabus policies and resources are provided here, https://www.cs.arizona.edu/cs-course-syllabus-policies :

- Department Code of Conduct
- Class Recordings
- Illnesses and Emergencies
- Obtaining Help
- Preferred Names and Pronouns
- Confidentiality of Student Records
- Additional Resources
- Land Acknowledgement Statement

## Subject to Change Statement

Information contained in the course syllabus, other than the grade and absence policy, may be subject to change with advance notice, as deemed appropriate by the instructor.