# C S 380I Structure and Implementation of Modern Programming Languages (SIMPL)

Welcome to C S 380I Structure and Implementation of Modern Programming Languages (SIMPL), the online master's offering for Fall 2026. Let's get started.

## 1 Personnel and Office Hours

The instructor is Dr. Siddhartha Chatterjee (sidchat@utexas.edu). The teaching assistants are Eric Harely, Conor McKenna, and Muhammed Mohaimin Sadiq.

The instructor holds office hours by appointment only over Zoom.

TAs will hold virtual office hours through Zoom according to a schedule posted on Canvas, and by appointment.

Course platforms: Canvas carries the gradebook, course materials, and announcements at https://utexas.instructure.com/courses/1462749; class discussion and Q&A run on Ed Discussion at https://edstem.org/us/courses/102175/discussion; programming assignments are submitted and auto-graded on Gradescope at https://utexas.instructure.com/courses/1462749/external_tools/151817.

## 2 Course Overview and Objectives

Programs written in high-level programming languages are the bridge between an algorithm's intent and the machine instructions realizing that intent. Driven by higher-level abstractions in modern programming languages and advances in instruction sets and micro-architectures, today's set of technologies for translating high-level program text to binary machine code have expanded significantly beyond the "classical" set of compilation technologies developed in the 1960s. This class samples the breadth of these technologies within a unified cost-benefit-risk analysis framework, and illustrates system integration by building a series of compilers for LiveOak, a pedagogical object-oriented language, targeting SaM, a simple abstract machine.

Core topics include the theory and practice of lexical analysis, syntactic analysis, code generation, language-level virtual machines, and dynamic memory management.

## 3 Prerequisites

The student should have experience with significant programming in Java, emphasizing the use, augmentation, and integration of pre-existing components.

Prior courses in programming languages, assembly language, compilers, computer architecture, and automata theory are helpful but not required.

## 4 Textbooks

None. Course materials — language specifications, starter code, and readings — are distributed through the course Canvas site.

## 5 List of Topics

The Fall 2026 semester runs Monday, August 24 through Monday, December 7, with the final exam in the university finals period (December 10–16). The topic sequence below carries the course's established arc across the sixteen-week semester. There is no class on Labor Day (Monday, September 7) or during the Thanksgiving break (Wednesday–Friday, November 25–27).

| Week | Dates | Topics | Assignments and Exams |
|---|---|---|---|
| 1 | Aug 24–28 | Course introduction. LiveOak, SaM. | SaM string library assigned. |
| 2 | Aug 31–Sep 4 | Practical parsing and code generation for SaM. | |
| 3 | Sep 8–11 | Practical parsing and code generation for SaM, continued. (Labor Day Mon 9/7.) | SaM string library due; LiveOak-2 to SaM compiler assigned. |
| 4 | Sep 14–18 | Theory of lexical analysis. | |
| 5 | Sep 21–25 | Theory of lexical analysis, continued. | |
| 6 | Sep 28–Oct 2 | Theory of syntactic analysis. | |
| 7 | Oct 5–9 | Theory of syntactic analysis, continued. | LiveOak-2 to SaM compiler due. |
| 8 | Oct 12–16 | No new topic; midterm week. | Midterm exam; scope through Week 7. |
| 9 | Oct 19–23 | Objects, classes, and inheritance. | |
| 10 | Oct 26–30 | Objects, classes, and inheritance, continued. | |
| 11 | Nov 2–6 | Implementing LiveOak-3. | LiveOak-3 to SaM compiler assigned. |
| 12 | Nov 9–13 | Implementing LiveOak-3, continued. Semantic analysis begins. | |
| 13 | Nov 16–20 | Semantic analysis. Type checking. | |
| 14 | Nov 23–24 | Memory management. Garbage collection. (Thanksgiving break Wed 11/25–Fri 11/27.) | |
| 15 | Nov 30–Dec 4 | Memory management. Garbage collection, continued. | LiveOak-3 to SaM compiler due. |
| 16 | Dec 7 | Course wrap-up. Last class day. | |
| | Dec 10–16 | University finals period. | Final exam. |

## 6 Assignments

There will be three programming assignments — the SaM string library, the LiveOak-2 to SaM compiler, and the LiveOak-3 to SaM compiler — ranging in duration from two to five weeks. All programming assignments must be attempted and completed individually. The programming language for the assignments is Java. Assignments will be submitted on-line for grading, and will be auto-graded. Repeated submissions are allowed and encouraged. The date of the last submission will be considered for computing slack days. Plagiarism-detection software will be used to flag possible violations of academic integrity.

Additionally, several *optional* problem sets will be handed out. These are intended solely to improve your understanding of the subject matter and help you prepare for the exams. You may work on them individually or in groups. The answer key for an individual problem set will be released two hours after the problem set is due.

## 7 Exams

There will be two timed on-line exams: a mid-term and a final. Both exams will be open-book and open-notes. The mid-term exam will be held in Week 8 (October 12–16) with scope through Week 7, and the final exam during the university finals period (December 10–16). Each exam will be open for a 120-hour period to allow students flexibility in taking them, and must be attempted in a single sitting. Further details will be distributed via Canvas and Ed Discussion once they are finalized.

## 8 Grading Policy

- Programming assignments are worth 60% of your grade.
- Exams are worth 40% of your grade (20% for mid-term, 20% for final exam).
- Exam grades may be curved. Program grades will not be curved.
- Grade cutoffs are as follows: A: 90% or higher; B: 80%–89.9%; C: 70%–79.9%; D: 60%–69.9%; F: 59.9% or lower.
- Problem sets are voluntary and do not count towards your grade. However, evidence of attempting them may be used to adjust borderline grades upwards.

## 9 Program Grade Requirements

The online master's program is a 30-hour program: 9 required hours and 21 elective hours. Required courses need a grade of B- or higher; elective courses, C or higher. To graduate, all students must have a graduate GPA average of at least 3.00.

## 10 Late Policy

Five slack days will be available for the semester and may be applied (in units of full days, i.e., no fractions of days) towards programming projects. Individual projects may limit the maximum number of slack days that can be applied toward them.

## 11 Tech Requirements

- Laptop or personal computer with the following requirements:
  - OS: Windows 11, macOS 13 or later, or a current Linux distribution
  - At least 10GB of free disk space
  - A 64-bit multi-core processor
  - 8GB RAM or better
  - Internet connection: broadband, 10 Mbps download / 5 Mbps upload or better, sufficient for video streaming and Zoom
  - Browser: latest version of Google Chrome or Firefox (Chrome is preferred)
- A Java development environment (JDK and an editor or IDE of your choice) for the programming assignments.
- Smartphone or scanner to take pictures and make PDFs of homework submissions.
- Smartphone or other device capable of being used for dual-factor authentication.

## 12 Academic Integrity in This Online Course

The online course format allows for multiple methods of identity verification and monitoring/detection of collusion, collaboration, and plagiarism. A violation of the course policy may include (but is not limited to) the following:

- Providing your UT EID to any other person.
- Collaborating or sharing information with another person regarding the material on any quiz, assessment, or assignment, before, during, and/or after any quiz, assessment, or assignment.
- Recording any quiz, assessment, or assignment material in any format.
- Failing to properly cite language, ideas, data, or arguments that are not originally yours.
- The public (such that it can be viewed by more than one person) posting of any form of a test bank or group of questions from any assignment.
- Consulting forbidden materials or sources of information.

The University of Texas at Austin Academic Integrity principles call for students to avoid engaging in any form of academic dishonesty on behalf of yourself or another student. Grade-related penalties are routinely assessed ("F" in the course is not uncommon), but students can also be suspended or even permanently expelled from the University for scholastic dishonesty.

If you have any questions about what constitutes academic dishonesty, please refer to the Dean of Students website or contact the instructor for this course.

You must agree to abide by the Honor Code of the University of Texas. You will not work with or collaborate with others in any way while completing any of the graded course assignments.

Use of AI tools in the programming assignments is permitted. Your submission remains your individual work and your individual responsibility: you must understand, and be able to explain, everything you submit, whatever produced it, and the prohibition on collaborating with other people stands unchanged. The exams carry their own AI policy: permitted, with mandatory disclosure — your exam submission includes a short AI-use disclosure naming the tools used and on which questions.

## 13 Accommodations

The University of Texas at Austin guarantees that students with disabilities have access to appropriate accommodations. You may request an accommodation letter from Disability and Access (D&A).

If you have approved accommodations for the course, please contact us to arrange them. Please do this as soon as possible, so that you can have the benefit of the accommodations throughout the duration of the course.

## 14 Course Etiquette

We expect that you will treat online discussions as though you are having a civil, respectful discussion with your fellow classmates in the same classroom. Please refrain from using profanity or any euphemisms for profanity. Please do not bait other commenters or personally attack them. Please do not use sarcasm in a way that can be misinterpreted negatively. And please do not make the same point repeatedly. In short, please just respect the right of your colleagues to ask questions and discuss their opinions about the subject matter of our course on the discussion board. Violators of these discussion rules will simply be shut out from all class communications — email, Ed Discussion, and office hours.

## 15 Academic Advisor Support

If you have additional questions or require support from an academic advisor, please contact the program coordinator at msonline@cs.utexas.edu.