COS 301: Programming Languages

Sudarshan S. Chawathe

University of Maine

Fall 2025

THIS COURSE IS AN UNDERGRADUATE INTRODUCTION TO PROGRAMMING LANGUAGES. Its catalog description states:

> Introduction to the fundamental concepts of formal languages, as well as the evolution and characteristics of major programming languages. Topics include finite-state automata, regular expressions, parsing, syntax and semantics, scope and binding, data types, and abstract data types.

We study programming languages drawn from four paradigms: imperative, object-oriented, functional, and logic. Specific languages include (not exclusively) an assembly language (Python bytecode), C, C++, Python, Java, Standard ML, and Prolog. We will compare features and styles and learn a bit about implementation aspects. We adopt an active learning approach that emphasizes exploration.

Prerequisites: COS 226 (data structures); COS 250 (discrete structures) with a grade of C or higher; programming maturity.

News and Reminders:
- Some sections below point to material in separate documents that are found on the class Web site, linked from the online version of this document.
- The most recent version of this document may be found at `http://chaw.eip10.org/cos301/`.
- Please use the PDF version of this document for printing and reference: `cos301.pdf`
- Brightspace site (access limited): `https://courses.maine.edu/d2l/home/425880`

Goals and Outcomes

Goals

- Learn about the historical development of programming languages.
- Study different programming language paradigms, including imperative, object-oriented, functional, and logic.
- Gain some experience in programming in diverse languages spanning the above paradigms.
- Learn some aspects of implementing programming languages, such as scanning (tokenization), parsing, interpretation, compilation, and optimization.
- Study underlying formalisms and methods, such as regular expressions and context-free languages.
- Improve programming skills.
- Improve communication skills, with particular emphasis on written communication and, further, well-written programs.

Student Learning Outcomes

Upon successful completion of this course, students should be able to
- List the important historical developments related to programming languages.
- Express an algorithm in multiple programming languages from diverse paradigms.
- Effectively use the characteristic features of programming languages from diverse paradigms.
- Explain the benefits and limitations of different programming languages and paradigms for solving a given problem.

- Implement simple programming languages using well established tools and techniques.
- Explain the inner workings of tokenizers and parsers and the underlying theory of regular expressions, automata, context-free grammars, etc.
- Effectively read suitable publications related to the topic.
- Use resources such as others' code and writing in an ethical and professional manner.
- Contribute to the body of knowledge at an undergraduate level.
- Analyze the correctness and running times of programs using simple methods.
- Perform simple experimental studies of programs.
- Program with attention to community standards and good practices.
- Communicate their programming work effectively.
- Meet Quantitative Literacy General Education requirements, such as being able to [following text is from U. Maine Gen. Ed. documents]:
  - Translate problems from everyday spoken and written language to appropriate quantitative questions.
  - Interpret quantitative information from formulas, graphs, tables, schematics, simulations, and visualizations, and draw inferences from that information.
  - Solve problems using arithmetical, algebraic, geometrical, statistical, or computational methods.
  - Analyze answers to quantitative problems in order to determine reasonableness. Suggest alternative approaches if necessary.
  - Represent quantitative information symbolically, visually, and numerically.
  - Present quantitative results in context using everyday spoken and written language as well as using formulas, graphs, tables, schematics, simulations, and visualizations.

Contact Information

Class meetings:

> **Time:** Mondays, Wednesdays, and Fridays, 09:00–09:50 a.m.
> **Location:** Barrows Hall, Room 130.

Instructor: Sudarshan S. Chawathe

> **Office:** Boardman Hall, Room 329.
> **Office hours:** (Please check for changes.)
> > Mondays, Wednesdays, and Fridays; 10:00 a.m.–10:50 a.m. (An appointment is *not required*, but advance notification is *useful*.)
> > Others by appointment, possibly remote/online.
> **Phone:** +1-207-581-3930.
> > *Please avoid calling* except for truly urgent matters.
> **Email:** sudarshan.chawathe@maine.edu
> > Use email only for messages unsuitable for the discussion forum. (See below.) Please use only this email address and put the string *COS301* near the beginning of the Subject header of the message. *Responses to all other messages may be very significantly delayed.*
> **Web:** http://chaw.eip10.org/.

Teaching Assistant: Sepideh Neshatfar

> **Office hours:** (Please check for changes.) In Boardman Hall, Room 138.
> > - Mondays and Tuesdays: 3:00–5:00 p.m.
> **Email:** sepideh.neshatfar@maine.edu

Online Resources

Class Web site:
> http://chaw.eip10.org/cos301/

We will use this Web site as the primary Web site for this class, and for posting assignments, readings, notes, and other material. Please monitor it frequently.

Class discussion forum: We will use the University's *Brightspace* installation for class discussions outside class meeting times. Brightspace will be used only for a few purposes; the class Web site noted above is the primary Web resource.

Class mailing list: *Please make sure you are on the class mailing list.* The mailing list will use the email address for each student as recorded in the official university records (*MaineStreet* system). We will use this mailing list only for urgent messages because all other messages will go on the class discussion forum. I anticipate fewer than a dozen messages on this list over the semester.

Grading Scheme

Grade components: *Students are expected to complete and submit all assigned coursework in good faith; those who fail to do so will earn a* **failing grade, regardless of overall numerical score**. Cut-offs for final letter grades D, C, B, A are, respectively, 35, 55, 70, and 85. Actual grades may be higher; these cut-offs provide lower bounds.

component	% of grade
class participation & exercises	10
homeworks	20
two quizzes	20
two midterm exams	30
final exam	20

Attendance: Attending class is required. Each student is granted three penalty-free absences for the semester, no questions asked. Beyond those three, each absence results in a *loss of one overall grade percentage point.* Absences for valid reasons (e.g., medical, family, religious, academic, athletic) may be excused *only if* a request is made very soon after the circumstances are known.

Class participation: Students are expected to contribute to learning by asking questions and making relevant comments in class and participating in the specified online components of the class. Quality is more important than quantity. Disruptive activity contributes negatively. See policies below.

Classroom exercises: Our work in the classroom may include some short individual and/or group exercises, meant to solidify understanding of the concepts being discussed. The exercises will be graded primarily for effort, group work, and other contributions, and less so for simple correctness.

Homeworks: Homeworks may include programming and non-programming components. No collaboration is permitted. Everyone is encouraged to discuss the problems and solution strategies *at a high level*, but the final solution and details must be individual work. If the boundary between permissible and non-permissible interactions is unclear, please ask for clarifications.

Exams and quizzes: All exams and quizzes are *closed book*; books, notes, or other such materials are not permitted. The use of computers and similar devices (tablets, phones, etc.) during exams is also *not allowed*. However, each student is permitted to bring a single standard Letter-sized sheet of paper that has notes that they have hand-written themselves. Both sides of this sheet may be used. If used, such notes must be submitted with the completed exam. Not doing so is considered academic dishonesty. Communications of any sort (electronic or other) are *not allowed*, except for communication between a student and the exam proctor.

Policies

Due dates: All due dates and times, as announced in class, are strict, to the second. If you believe your work was delayed by truly exceptional circumstances, let me know as soon as those circumstances are known to you and I will try to make a fair allowance. However, *the default is that you get a zero if you don't turn in the work on time*, and fail the class if you don't turn it in at all (cf. Grade Components above).

Attendance: It is very important to attend all class meetings, for many reasons, but the grade component provides additional motivation to those who may need it. If you have a valid reason for missing a class, let me know early and I will try to help you make up the class. There will be no make-up exams or quizzes. A missed test earns zero credit. If you have a valid reason for missing a test, let me know as early as that reason is known to you and I will make a fair allowance (typically by scaling other scores) but *there will be no make-up tests in any case.*

Classroom activities: This course is based on an active learning format, so effective classroom activities are critical to its success. Students are expected to contribute to their own learning and to that of their classmates, and to devote 100% of their attention to these activities while in class. For this reason, electronic and other distractions (computers, phones, assorted gizmos, etc.) may be required to be be completely silenced and put away for some or all of the duration of class meetings. (Students who need any such devices for disability accommodations should follow the appropriate procedures, or contact the instructor if in doubt. Others who need any accommodation in this regard due to special circumstances should make advance arrangements with the instructor.) No food or drink is allowed in class, other than water, tea, coffee, and similar, in a spill-proof container. (As a general guideline, we will follow "library rules" in this regard.) Students who violate these rules or otherwise cause distractions in class will be asked to leave with *no warning*; habitual violators will face disciplinary action.

Office hours: All students are encouraged to make use of office hours to further their learning, obtain assistance on homework assignments, obtain feedback on their class performance, etc. However, office hours are not to be used as a substitute for attending and participating in class meetings (see above). Similarly, assistance with homework assignments will be limited to what is appropriate based on fairness to all; students are expected to demonstrate substantial effort on the assignment before seeking assistance.

Make-up classes: I may have to reschedule a few classes due to my other professional commitments. I will make every attempt to minimize the number of such occurrences and to reschedule for a time that works for most students. Further, I will make sure no student is penalized by such occurrences.

Use of Generative AI tools: Among the very many available resources that may be used to assist with coursework (books, Web sites, tutorial videos, etc.), recent and rapidly evolving generative AI (GenAI) tools, such as ChatGPT, perplexity.ai, and many more, perhaps deserve a separate clarification here.

Briefly, the use of such GenAI tools is *allowed **subject to important conditions** (below), but not recommended.*

If GenAI tools are used then:

- The use must be properly cited in a prominent manner in all submitted material (in particular including code). The format suggested by the following example must be used:

  "Write a regular expression that recognizes Java numeric literals" prompt to ChatGPT. 2025-09-12 04:45. `https://chatgpt.com/`.

- All parts of the submitted material that were generated by such tools (even if further edited) must be very clearly delineated. For example, if a regular expression (or other code snippets) resulting from the above prompt is used in submitted code, that use must be very clearly marked using both inline comments in the code and comments in the accompanying documentation (such as a README file).

- The submitter must understand all submitted material, regardless of source. In particular, this requirement means that everyone must be able to explain, in detail, all their submitted code if and when they are called upon to do so. In the ongoing example, the submitter must be able to explain exactly how the regular expression works, why it is needed, etc.

- The submitter is responsible for all submitted content, regardless of source. For instance, in the ongoing example, if the code snippet related to regular expression contains a benign or malicious bug that does horrible things, the submitter is responsible in the same manner as if that code were written without such tools.

It is the submitter's responsibility to ask for clarifications and ensure compliance with these requirements. Failure to do so is a serious case of academic dishonesty and will be subject to the usual (severe) penalties.

University of Maine administrative policy statements: [Verbatim, standard wording from `https://umaine.edu/citl/teaching-resources-2/required-syllabus-information/`. Please refer to that site for further details.]

Academic Honesty Statement Academic honesty is very important. It is dishonest to cheat on exams, to copy term papers, to submit papers written by another person, to fake experimental results, or to copy or reword parts of books or articles into your own papers without appropriately citing the source. Students committing or aiding in any of these violations may be given failing grades for an assignment or for an entire course, at the discretion of the instructor. In addition to any academic action taken by an instructor, these violations are also subject to action under the University of Maine Student Conduct Code. The maximum possible sanction under the student conduct code is dismissal from the University.

Students Accessibility Services Statement If you have a disability for which you may be requesting an accommodation, please contact Student Accessibility Services, 121 East Annex, 581.2319, as early as possible in the term. Students who have already been approved for accommodations by SAS and have a current accommodation letter should meet with me (the instructor of the course) privately as soon as possible.

Course Schedule Disclaimer (Disruption Clause) In the event of an extended disruption of normal classroom activities, the format for this course may be modified to enable its completion within its programmed time frame. In that event, you will be provided an addendum to the syllabus that will supersede this version.

Observance of Religious Holidays/Events The University of Maine recognizes that when students are observing significant religious holidays, some may be unable to attend classes or labs, study, take tests, or work on other assignments. If they provide adequate notice (at least one week and longer if at all possible), these students are allowed to make up course requirements as long as this effort does not create an unreasonable burden upon the instructor, department or University. At the discretion of the instructor, such coursework could be due before or after the examination or assignment. No adverse or prejudicial effects shall result to a student's grade for the examination, study, or course requirement on the day of religious observance. The student shall not be marked absent from the class due to observing a significant religious holiday. In the case of an internship or clinical, students should refer to the applicable policy in place by the employer or site.

Sexual Violence Policy Sexual Discrimination Reporting
The University of Maine is committed to making campus a safe place for students. Because of this commitment, if you tell a teacher about an experience of sexual assault, sexual harassment, stalking, relationship abuse (dating violence and domestic violence), sexual misconduct or any form of gender discrimination involving members of the campus, your teacher is required to report this information to the campus Office of Sexual Assault & Violence Prevention or the Office of Equal Opportunity.
If you want to talk in confidence to someone about an experience of sexual discrimination, please contact these resources:
For confidential resources on campus: Counseling Center: 207-581-1392 or Cutler Health Center: at 207-581-4000.
For confidential resources off campus: Rape Response Services: 1-800-310-0000 or Partners for Peace: 1-800-863-9909.
Other resources: The resources listed below can offer support but may have to report the incident to others who can help:
For support services on campus: Office of Sexual Assault & Violence Prevention: 207-581-1406, Office of Community Standards: 207-581-1409, University of Maine Police: 207-581-4040 or 911. Or see the OSAVP website for a complete list of services at http://www.umaine.edu/osavp/

Programming

This is a course on programming languages and will cover a few in detail and more in less detail, from a list that includes C, C++, Python, Java, Standard ML, and Prolog. **The only fully supported platform is Debian (stable) GNU/Linux.** (In case of a new major release very close to the semester's start, oldstable will be supported instead of or in addition to stable; details should be clarified in class.) Therefore,

everyone should ensure that they have such a machine (real or virtual) that they own and fully control. Most of what we need is likely to work on a variety of other platforms too but support for those will be increasingly limited with increasing difference from the main platform. (For example, Debian derivatives like Ubuntu will almost certainly be fine; Windows, MacOS much less so.) The reason is simple: We wish to spend more time on the core topics and less on debugging arcane systems issues related to quirks of a platform or installation. Submissions will be in the form of packaged, well documented *source* files. *Proper documentation and packaging of source code and other material is a crucial component of assigned work and submissions failing in this regard will receive no credit.*

Literate Programming: All submitted work must use a *literate programming style*: Your programs must be designed with *a human as the intended reader*, although they must also compile and run correctly. *Programs that do not meet this requirement are likely to receive a zero score with no further consideration.* Details will be discussed in class.

Schedule

A rigid schedule is not conducive to effective learning, since it would limit our flexibility in exploring ideas as they arise in class. A partial and *approximate* schedule, to serve as a baseline, appears in Figure 1; it will be updated as we progress. Please use it only as a rough guide to plan your studies. *Do not use it to schedule travel or other events.* If you need a definite answer on when something will or will not occur, you should check with me.

At the beginning and end of each class, I typically announce the topics and textbook sections covered in that class and those due at the next class. It is important that students read the material *before* the class in which it is discussed and, in general, keep up with readings and studies.

Textbook and Readings

Textbook: Kent D. Lee. *Foundations of Programming Languages.* Undergraduate Topics in Computer Science. Springer Nature, 2nd edition, December10 2017. *Please note the edition and year.* It is a *required textbook* for this course. *The ebook version is available gratis via the University library.*

There are some resources on the Web for this textbook and more generally for this material. You are welcome, and encouraged, to use these resources (unless specifically directed otherwise), and to share and discuss them with classmates on the discussion forum. However, *you must prominently attribute any help from such or other resources in all your work.* Failure to do so is a serious offense (see policies). And, regardless of what resources you use or do not, it is never permissible to simply include someone else's work unless it is specifically permitted as an exception.

Readings: Additional readings may be added here based on class needs and preferences as the semester progresses. Unless indicated otherwise, all readings are *required*.

1. David Beazley. PLY (Python Lex-Yacc). `http://www.dabeaz.com/ply/`, February 2020.

2. Kent D. Lee. The JCoCo virtual machine 1.0 documentation. `https://kentdlee.github.io/JCoCo Pages/_build/html/index.html`, 2017.

Exercises, Homeworks, Tests, and Notes

It may be useful to refer to material from the previous sessions (recursively): `http://chaw.eip10.org/202501/cos301/`. However please bear in mind that each session is different based on a variety of factors. Additional material will appear here as we move along the semester.

- Presentation slides from class. Caution: May be misleading without appropriate context of accompanying classroom discussion, and definitely do not include many important discussions that arise organically.

Monday	Wednesday	Friday
September 1st ✕ *No class.* Labor Day.	3rd C1 Introduction. § 1.4.	5th C2 Syllabus, § 1.4.
8th C3 §§ 2.0–2.1.	10th C4 §§ 2.2–2.3. HW01 out.	12th C5 §§ 2.4–2.5.
15th C6 § 2.6.	17th C7 *PLY* (see readings). HW01 due.	19th C8 § 2.7, PLY. HW02 out.
22nd C9 §§ 2.8-2.9.	24th C10 § 2.10.	26th C11 ★ **Quiz 1**, regular class time & place. Catch-up, review.
29th C12	**October** 1st C13 §§ 2.11–2.$. HW02 due.	3rd C14 §§ 3.0-3.1. *JCoCo* (see readings).
6th C15 §§ 3.2-3.3. HW03 out.	8th C16 ★ **Midterm Exam 1**, regular class time & place. Catch-up, review.	10th C17
13th ✕ *No class.* Fall Break.	15th C18 §§ 3.4-3.5.	17th C19 §§ 3.6-3.7
20th C20 §§ 3.6-3.7	22nd C21 §§ 3.8-3.9	24th C22 §§ 3.10-3.12 HW03 due.
27th C23 §§ 3.12-3.13	29th C24 §§ 5.1,5.3,5.4. HW04 out.	31st C25 §§ 5.5
November 3rd C26 §§ 5.6,5.7,5,8	5th C27 §§ 5.9,5.10	7th C28 ★ **Quiz 2**, regular class time & place. Catch-up, review.
10th C29 §§ 5.11,5.12,5.13	12th C30 §§ 5.14,5.15,5.16	14th C31 §§ 5.* HW04 due.
17th C32 §§ 7.0–7.3	19th C33 ★ **Midterm Exam 2**, regular class time & place. Catch-up, review.	21st C34 §§ 7.4–7.5 HW05 out.
24th C35 §§ 7.6–7.7	26th ✕ *No class.* Thanksgiving Break.	28th ✕ *No class.* Thanksgiving Break.
December 1st C36 §§ 7.8–7.9	3rd C37 catch-up; review.	5th C38 catch-up; review. HW05 due.
8th C39 Catch-up, synthesis, and review.	10th C40 Catch-up, synthesis, and review.	12th C41 Catch-up, synthesis, and review.
15th ✕ *No class.* ★ **Finals week.** Check Univ. schedule.	17th ✕ *No class.* ★ **Final exam: 12:15pm–2:15pm; WW 130.**	19th ✕ *No class.* ★ **Finals week.**

Figure 1: **Approximate** schedule, likely to change. Notation: **§§ *x.y*** ⇒ textbook chapter *x*, section *y*.

- P01: `pres/pl01pres.pdf`.
- P02: `pres/pl02pres.pdf`.
- P03: `pres/pl03pres.pdf`.
- P04: `pres/pl04pres.pdf`.
- P05: `pres/pl05pres.pdf`.
- P06: `pres/pl06pres.pdf`.
- P07: `pres/pl07pres.pdf`.
- P08: `pres/pl08pres.pdf`.
- P09: `pres/pl09pres.pdf`.

- Class exercises:

  - Class Exercise 1: `hwq/ce01.pdf`.
  - Class Exercise 2: `hwq/ce02.pdf`.
  - Class Exercise 3: `hwq/ce03.pdf`.

- Homework assignments:

  - Homework 1: `hwq/hw01.pdf`.
  - Homework 2: `hwq/hw02.pdf`.
  - Homework 3: `hwq/hw03.pdf`.

- Quizzes and Exams:

  - Quiz 1:
    * Quiz 1: `hwq/q01.pdf`.
    * Sample solutions: `p/q01s.pdf`.
  - Midterm Exam 1:
    * Midterm Exam 1: `hwq/mt01.pdf`.
    * Sample solutions: `p/mt01s.pdf`.
  - Quiz 2:
    * Quiz 2: `hwq/q02.pdf`.
    * Sample solutions: `p/q02s.pdf`.
  - Midterm Exam 2:
    * Midterm Exam 2: `hwq/mt02.pdf`.
    * Sample solutions: `p/mt02s.pdf`.

Homework and Project Submissions

All electronic submissions must be made using the procedure that will be outlined in class. Electronic submissions in **all other forms**, such as email or physical media, will be **discarded and receive no credit.**
Illegible, hard to read, or otherwise messy submissions, whether handwritten on typed, are likely to be returned without grading, for zero credit.

Fallback procedure If (and only if) there are unexpected problems and you are unable to submit your work as above, then you should save your file on your own computer (with some backups), compute its MD5 checksum using the md5sum utility on Unix-like systems (or other similar tools), and submit the file name, time stamp, and MD5 checksum (only, not the file itself) by email with a suitable Subject header.