



CSE 6341: Foundations of Programming Languages

Instructor: Atanas (Nasko) Rountev, email: rountev.1@osu.edu

Office hours: see Carmen; Grader: see Carmen

Course overview

Course description

This is a course on the theory of programming languages. The goal is to study formal ways of defining the syntax and semantics of programming languages. The main topics are attribute grammars, type systems, operational semantics, abstract interpretation, and language implementation via interpreters and compilers. This is a rather theoretical course that requires understanding and applying various formalisms in the context of programming languages. To balance this theoretical material, the coursework includes substantial programming projects that demonstrate the actual implementation of such syntax and semantics definitions.

Prerequisites

CSE 3341/5341: Principles of Programming Languages. For reference, the official syllabus for this prerequisite course is provided at the 6341 web page under “Resources”. An equivalent undergraduate programming language is an acceptable substitute.

CSE 6341 is a course from the graduate foundational core and the level of difficulty is relatively high. If you are an undergraduate student with good math abilities, programming skills, and willingness to do graduate-level work, you should do fine; if not, you should probably not take this course. If you are a graduate student in another department, you must have in-depth experience with imperative and object-oriented programming, and background in formal languages and grammars.

Learning outcomes

By the end of this course, students should successfully be able to:

- Master using attribute grammars to analyze standard imperative language constructs
- Master the implementation of aspects of programming language interpreters and compilers
- Be competent with analyzing types in imperative and object-oriented languages
- Be competent with using attribute grammars to specify code generation for standard imperative language constructs
- Be competent with analyzing the formal semantics of imperative languages
- Be competent with abstract interpretation and its application to derive approximate semantics of programs

Office hours

The instructor’s office hours will be held in person (once a week) and via Zoom (once a week).

The grader’s office hours will be held similarly. See Carmen for details. If you cannot make it during the scheduled office hours, please arrange an alternative meeting time via email.

Course materials

Textbooks

This course does not have a required textbook. We will use materials from several books, as described below, but these will be optional. Your most important reading will be the lecture notes and your own notes. Copies of the lecture notes will be available on the course web page, organized by topic. See the course web page under “Resources” for details on these textbooks.

- Pagan, Formal Specification of Programming Languages: A Panoramic Primer, 1981
- Slonneger and Kurtz, Formal Syntax and Semantics of Programming Languages, 1995
- Aho, Lam, Sethi, Ullman, Compilers: Principles, Techniques, and Tools, 2nd Edition (also called the “Dragon Book”), 2007
- Nielson and Nielson, Semantics with Applications: A Formal Introduction, 1992

Written assignments, programming projects, exams

Written assignments

- There will be several written assignments. They will be posted via Carmen and your submissions will be uploaded via Carmen.
- Assignments should be done independently. General high-level discussion of assignments with other students in the class is allowed, but **all actual work should be your own**. Written assignments that show excessive similarities will be taken as evidence of cheating and dealt with accordingly. See more details below under “Academic integrity”.
- Written assignments should be turned in by 11:59 pm on the due day. You can submit up to 24 hours after the deadline; if you do so, your score will be reduced by 10%.
- **If you submit more than 24 hours after the deadline, the submission will not be accepted. NO EXCEPTIONS TO THIS RULE WILL BE CONSIDERED. NO REQUESTS FOR RESUBMISSION WILL BE CONSIDERED.**

Programming projects

- There will be several programming projects. They will be posted via Carmen. Your submissions must be uploaded via Carmen by midnight on the due date.
- The projects must compile and run on **stdlinux**. Some students prefer to implement the projects on a different machine, and then port them to stdlinux. If you decide to use a different machine, it is entirely your responsibility to make the code compile and run correctly on stdlinux before the deadline. In the past many students have tried to port to stdlinux too close to the deadline, leading to last-minute problems and missed deadlines.
- Projects should be done independently. General high-level discussion of projects with other students in the class is allowed, but **you must do all design, programming, testing, and debugging independently**. Projects that show excessive similarities will be taken as evidence of cheating and dealt with accordingly. Code plagiarism tools may be used to detect cheating. See more details below under “Academic integrity”.
- The projects are due by 11:59 pm on the due day. You can submit up to 24 hours after the deadline; if you do so, your score will be reduced by 10%. **ONLY THE LAST SUBMITTED VERSION WILL BE CONSIDERED.** Triple-check carefully that you have submitted the correct version. **If you submit the wrong version of your code, and you get a low score (or zero score), I will NOT consider resubmissions – the original low/zero score will be assigned WITHOUT DISCUSSION.**

- **If you submit more than 24 hours after the deadline, the submission will not be accepted. NO EXCEPTIONS TO THIS RULE WILL BE CONSIDERED. NO REQUESTS FOR RESUBMISSION WILL BE CONSIDERED. MAKE SURE YOU SUBMIT THE CORRECT CODE VERSION.**
- Read the project description **very carefully, several times, start-to-end**. If you need any clarifications, contact me immediately (do **not** wait until the last minute). **Test extensively**.

Exams

There will be a midterm exam and a final exam.

- The midterm exam will be on **February 22 (Thursday)** during class time.
- The final exam will be on **April 29 (Monday), 8:00 am – 9:45 am**.
- Both exams will be in person in the classroom.
- You must complete the midterm and final exams yourself, without any external help or communication. See more details below under “Academic integrity”.
- Both exams will be comprehensive (i.e., cover all material from the start of the semester up to and including the last lecture before the exam). **The exams are open book/notes/computer. You can access the Internet and local files on your device. No communication with anyone during the exam is allowed, in person or online (no emails, text messages, chats, etc.)**
- Exam questions will require creative application of approaches discussed in class. Memorizing things will not be enough; you need to have conceptual understanding of the techniques we have discussed, and how these techniques could be applied to small problems. Exam questions will be very similar to the questions from the written assignments; thus, you should make sure that you have solid understanding of all details in the solutions for written assignments.
- Missing the midterm or the final without prior written (e-mail) approval from me will result in a score of zero for that exam. To get my approval to reschedule an exam, e-mail me at least one week before the exam is scheduled. I will not give approval unless the reasons are justifiable (typically, medical or family emergencies).

Grading

Written assignments: **15%**; programming projects: **40%**; midterm exam: **15%**; final exam **30%**. The course will be graded on a curve. I expect the median grade to be a bit above B+. Statistics will be provided to help you understand your standing in the class. **Note that Carmen is not set up to reflect this grading scheme - it is only used to show the raw scores for programming projects and written assignments. Ignore anything in Carmen that shows cumulative scores/percentages or overall course grades, since they are meaningless.**

I will grade the midterm and the final. The grader will grade the written assignments and the programming projects. The person who graded something will be responsible for handling grading disputes. A grade becomes final a week after being handed back. This should leave enough time to resolve grading disputes. If there are unforeseen emergencies that affect the planned grading scheme, appropriate adjustments will be made. I will provide as much advance notification of such changes as possible under the circumstances.

Other course policies

Discussion and communication guidelines

The following are my expectations for how we should communicate as a class. Above all, please remember to be respectful and thoughtful.

- **Writing style:** a common theme of this course is the application of theoretical principles to problems in programming languages. As with all theoretical foundations, your solutions must be **precise and detailed:** you must work out all details that are necessary to solve the problem using the approaches discussed in class. You must write your solutions in a way that convinces the reader that you understand all these details. Be careful, precise, and thorough.
- **Tone and civility:** maintain a supportive learning community where everyone feels safe and where people can disagree amicably and professionally. I am committed to making the classroom a comfortable space for all of us. We will respect each other and practice civility at all times. Disrespectful language including, but not limited to, sexist, racist, homophobic, or anti-ethnic slurs, or bigotry will not be tolerated.

Academic integrity policy

- **Midterm exam and final exam:** You must complete the midterm and final exams yourself, **without any external help or communication** (but you can ask me clarifying questions during the exams).
- **Written assignments:** Your written assignments should be your own original work. General high-level discussion of assignments with other students in the class is allowed, but all actual work should be your own. **Do not provide your own solutions to other students and do not look at other students' solutions.**
- **Programming projects:** Projects should be done independently. General high-level discussion of projects with other students in the class is allowed, but **you must do all design, programming, testing, and debugging independently. Do not provide your own solutions to other students and do not look at other students' solutions.** Code plagiarism tools may be used to detect cheating.
- **Reusing past work by you or others:** **You are prohibited from turning in work (by you or others) from a past class to your current class, even if you modify it.** If you want to build on past work or revisit a topic you have explored in previous courses, please discuss the situation with me.

General information not specific to this course

COVID

All students, faculty, and staff are required to comply with and stay up to date on all university safety and health guidance (<https://safeandhealthy.osu.edu/>). Accommodations will be made based on university guidelines.

Ohio State University's academic integrity policy

It is the responsibility of the Committee on Academic Misconduct ([COAM Home](#)) to investigate or establish procedures for the investigation of all reported cases of student academic misconduct. The term "academic misconduct" includes all forms of student academic misconduct wherever committed; illustrated by, but not limited to, plagiarism, collusion (unauthorized collaboration), copying the work of another student, and possession of unauthorized materials during an examination. Instructors shall report all instances of alleged academic misconduct to the committee (Faculty Rule 3335-5-487). For additional information, see the Code of Student Conduct <http://studentlife.osu.edu/csc/>.

If I suspect that a student has committed academic misconduct in this course, I am obligated by university rules to report my suspicions to the Committee on Academic Misconduct. If COAM determines that you have violated the university's *Code of Student Conduct* (i.e., committed academic misconduct), the sanctions for the misconduct could include a failing grade in this course and suspension or dismissal from the university.

If you have any questions about the above policy or what constitutes academic misconduct in this course, please contact me. Other sources of information on academic misconduct (integrity) to which you can refer include:

- Committee on Academic Misconduct web pages ([COAM Home](#))
- Ten Suggestions for Preserving Academic Integrity ([Ten Suggestions](#))
- AI and Academic Integrity (<https://oaa.osu.edu/artificial-intelligence-and-academic-integrity>)
- Avoiding Plagiarism (<https://research.osu.edu/research-responsibilities-and-compliance/research-misconduct/avoiding-plagiarism>)

Statement on Title IX

All students and employees at Ohio State have the right to work and learn in an environment free from harassment and discrimination based on sex or gender, and the university can arrange interim measures, provide support resources, and explain investigation options, including referral to confidential resources.

If you or someone you know has been harassed or discriminated against based on your sex or gender, including sexual harassment, sexual assault, relationship violence, stalking, or sexual exploitation, you may find information about your rights and options at titleix.osu.edu or by contacting the Ohio State Title IX Coordinator at titleix@osu.edu. Title IX is part of the Office of Institutional Equity (OIE) at Ohio State, which responds to all bias-motivated incidents of harassment and discrimination, such as race, religion, national origin and disability. For more information on OIE, visit equity.osu.edu or email equity@osu.edu.

Accessibility accommodations for students with disabilities

The university strives to make all learning experiences as accessible as possible. If you anticipate or experience academic barriers based on your disability (including mental health, chronic, or temporary medical conditions), please let me know immediately so that we can privately discuss options. To establish reasonable accommodations, I may request that you register with Student Life Disability Services. After registration, make arrangements with me as soon as possible to discuss your accommodations so that they may be implemented in a timely fashion. SLDS contact information: slds@osu.edu; 614-292-3307; slds.osu.edu; 098 Baker Hall, 113 W. 12th Avenue.

Religious accommodations

It is Ohio State's policy to reasonably accommodate the sincerely held religious beliefs and practices of all students. The policy permits a student to be absent for up to three days each academic semester for reasons of faith or religious or spiritual belief. Students planning to use religious beliefs or practices accommodations for course requirements must inform the instructor in writing no later than 14 days after the course begins. The instructor is then responsible for scheduling an alternative time and date for the course requirement, which may be before or after the original time and date of the course requirement. These alternative accommodations will remain confidential. It is the student's responsibility to ensure that all course assignments are completed.