Course Summary

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 and their uses, axiomatic semantics, and operational semantics. In addition to these topics, we will cover some material on functional programming and type systems. This is a fairly theoretical course, and it requires understanding and applying various formalisms in the context of programming languages. To balance this theoretical material, the coursework includes a substantial number of 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 under “Resources” at the 6341 web page.

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.

Objectives

• Understand the role of certain theoretical formalisms, and apply them in the context of programming languages.
• Use attribute grammars to specify context-sensitive conditions, compile-time analyses, and translational semantics.
• Define the axiomatic semantics of simple imperative constructs, and use it to prove program properties.
• Define the operational semantics of simple imperative languages and functional languages.
• Use type systems to specify compile-time properties and analyses.
• Implement parts of programming language interpreters and compilers.

Exclusions

Not open to students with credit for CSE 755

General Information

• Class time and location: Tuesday and Thursday, 11:10 am – 12:30 pm, Dreese 305
• Instructor: Atanas (Nasko) Rountev, rountev@cse.ohio-state.edu; office hours: DL 685, Tuesday and Thursday 12 pm – 1 pm, or by appointment
• Grader: Yang Xia, xia.425@osu.edu, Office hours: Dreese 190, Monday and Friday 10 am – 11 am, or by appointment

Reading

This course does not have a required textbook. We will use materials from several different books. See “Resources” on the course web page for details.


Lisp 1.5 Programmer’s Manual, McCarthy and others.

Your most important reading will be the lecture notes and your own notes. Copies of all notes will be handed out in class, and will also be available on the course web page. For each topic, I will give you pointers to relevant parts of the books.

**Course Web Page**

http://www.cse.ohio-state.edu/~rountev/6341: the course web page will contain all notes, handouts, assignments, a detailed schedule, pointers to reading materials, etc.

**Piazza Discussions**

We will use Piazza for questions and discussions. Sign up at piazza.com/osu/autumn2017/cse6341. If you have a question, it is highly preferable to post it to Piazza instead of emailing me directly. If you prefer, you can post anonymously. When a question is posted and answered on Piazza, it benefits everyone.

**Assignments**

- There will be several written assignments, typically due in 7 days.
- 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. Assignments that show excessive similarities will be taken as evidence of cheating and dealt with accordingly.
- Assignments should be turned in by the end of class on the due day. Late assignments turned in by the end of the next class will be graded with 30% reduction. Assignments turned in later than that will not be accepted.
- Make the assignments readable and understandable. They should be handed in on regular paper, legibly written or typed. If you have more than one sheet, please staple the sheets together.
- **Important:** A common theme of this course is the application of theoretical principles to problems in the domain of programming languages. As with all theoretical foundations, your solutions have to be precise and detailed: you have to work out all details that are necessary to solve the problem using the approaches discussed in class. You also have to write your solutions in a way that convinces the grader that you understand all these details. Be careful, precise, and thorough.

**Projects**

- There will be several programming projects, which have to be submitted electronically on stdlinux 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 their code 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 have to do all design, programming, testing, and debugging independently. Projects that show excessive similarities will be taken as evidence of cheating and dealt with accordingly.
- The projects are due by 11:59 pm on the due day. No exceptions will be made to this deadline: if you submit at 12:00 am, your submission will be considered to be late. Please plan your time carefully and do not submit in the last minute. The time stamp on the electronic submission will be used to determine the submission time. A reduction of 10% per day will be applied to late submissions. Submissions more than three days late will not be accepted.
Exams

• There will be a midterm exam and a final exam.
  – Midterm: October 10, Dreese 305
  – Final: December 11, 12 pm - 1:45 pm, Dreese 305
Both will be comprehensive, in-class, closed book. You will be allowed to use a cheat sheet: one letter-size piece of paper, with notes on both sides.

• The exam questions will typically require creative application of general approaches discussed in class. Memorizing things will not be enough; you need to have conceptual understanding of the techniques we have covered, and how these techniques could be applied to small problems. Exam questions will be very similar to the questions from the homeworks; thus, you should make sure that you have solid understanding of all details in the homework solutions.

• 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 such approval unless the reasons are justifiable.

Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>15%</td>
</tr>
<tr>
<td>Projects</td>
<td>50%</td>
</tr>
<tr>
<td>Midterm</td>
<td>10%</td>
</tr>
<tr>
<td>Final</td>
<td>25%</td>
</tr>
</tbody>
</table>

Grading Policy

The entire course will be graded on a curve. I expect the median grade to be around B+. Statistics will be provided to help you understand your standing in the class. I will grade the midterm and the final. The grader will grade the assignments and the programming projects. The person who graded something will be responsible for handling grading disputes. A grade become final one week after being handed back. This should leave plenty of 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.

Academic Integrity

I will treat you as professionals, and you should conduct yourselves as such. You are free to discuss the projects with others. However, the solutions you submit should be developed entirely by yourself. **Cheating is a very serious offense and will not be tolerated.** Supplying others with materials is also against this rule. Additional details on academic integrity are available at oaa.osu.edu/coamresources.html. Please read this information carefully.

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.