Contact Information

| Instructor: Dr. John C. Lusth | Email: lusth@cs.ua.edu | Web: http://beastie.cs.ua.edu/proglan/ |
| Office: 3427 SEC | Hours: signup sheet is on my office door | Phone: 205-348-1618 |

Catalog Statement

Formal study of programming language specification, analysis, implementation, and run-time support structures; organization of programming languages with emphasis on language constructs and mechanisms; and study of non-procedural programming paradigms.

Implementing the Catalog Statement

Students will embed features from various languages into Scam, a lexically-scoped variant of LISP. Students will also individually implement a programming language of their own design and write a significant program demonstrating the language.

Additional and orthogonal subjects to be covered in this class are: lexical analysis, syntactical analysis, parameter passing strategies, and axiomatic semantics.

At the end of this class, a student should be understand the concepts of: syntax vs. semantics, data vs. procedures, objects and state, functional programming, evaluation strategies, and interpretation vs. compilation.

Prerequisite

CS200 and CS201

Textbook and Supplies


A Linux-based laptop computer is required for this course. A Windows-10 installation that includes Bash is acceptable. A Mac installation is possible with some work.

Larger Issues

This class is not only concerned with what you learn, but how fast you learn. The only way to learn quickly is read in advance, take your study group seriously, ask copious questions, and start all programming tasks early.

Topics

The schedule for the class can be found at:

http://beastie.cs.ua.edu/proglan/schedule.html

Make sure you read the material assigned in class, work any exercises, and explore answering any questions prior to class.

Attendance

Attendance is not required. If you miss, you are expected to get any missing information from other members of the class, especially your study group. I will not tell you what you’ve missed as I am too lazy to lecture over the same material more than once.

An absence will be excused if you provide a written, university authorized, excuse.
Grading
Grades will be determined by the following percentages:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>Exam 1</td>
</tr>
<tr>
<td>20%</td>
<td>Exam 2</td>
</tr>
<tr>
<td>30%</td>
<td>Large programming assignment</td>
</tr>
<tr>
<td>30%</td>
<td>Small programming assignments (3)</td>
</tr>
</tbody>
</table>

There is no final exam in this course. The large programming assignment, the Designer Programming Language has milestones (as listed in the schedule).

Grading Methodology

For your designer programming language, you should implement the following support modules in the order given:

- Grammar
- Lexical Analyzer
- Parser
- Environment

The dates for having these milestones completed are found in the schedule. Failure to submit a milestone or submitting a very substandard milestone will result in a one point deduction from your final percentage. Submitting a partially working or semi-acceptable milestone will result in a half point deduction.

Your final project must have a certain level of functionality (given in the project description). Otherwise, you will receive a failing grade for the class.

For the small programming assignments, you must pass all tests. If you fail even one test, you will have to resubmit. You are guaranteed at least four submissions per assignment, with each failed submission resulting in a maximum deduction of 10 points (it may be less). The time between submissions will minimum be a minimum of two days, but usually weekly. I will announce in class the next resubmission date and deduction. Failure to submit by a resubmission deadline will be considered a submission that failed all tests. After the last allowed submission, you will receive a score of 25% if you can pass a majority of the tests. Otherwise, you will receive a zero.

For programming assignments, you will recieve a bonus based upon how successful your initial submission was. The formula for the bonus is the percentage of tests passed divided by twenty. For example, suppose you passed all tests on your third submission. In this case, your score for the assignment would be $100 - 10 - 10 + \frac{80}{20}$ or 84 out of 100. Passing all tests with the original submission will yield a score of 105 out of 100.

These are the kinds of things for which I won’t regrade an assignment submission:

- something incorrect that was pointed out as incorrect in a previous submission
- something incorrect that was shown how to be made correct in a previous submission
- something incorrect that the test dropbox would have pointed out as incorrect
- something incorrect that was shown how to be made correct in a task description
- your algorithm is incorrect
- other things in a similar vein

Otherwise, these are the kinds of things for which I may perform a regrade:

- you accidentally left in some debug code
- you accidentally deleted some code that was in a previous submission
- other things in a similar vein

These are the kinds of things for which I will most assuredly perform a regrade:

- your result is actually correct and mine is wrong
- your reasonable interpretation of a task description differs from my interpretation

Undergraduate students will be able to omit a certain number of questions on exams. Graduate students will not be able to omit any questions.

Grading is straight scale with the following cut-offs: 90% and above is an A, 80% and above is a B, 70% and above is a C, 60% and above is a D, and anything lower than 60% is an F. These cut-offs are strict. For example, 79.99999% is a C. The delta for plus-minus is two points. For example, 90..92 will be an A-, while 98..100 will be an A+.
Other Details

The following conventions will be followed in *Programming Languages*. Please make sure that you fully understand each of the items listed below.

- Attendance is not required. However, I generally will not answer questions that I have already answered in class. Use the forum (see below) to ask questions about material you missed from being absent.

- There are no make-up exams (unless you have a written, university-authorized excuse). If you miss an exam, a zero will be recorded for your score. You have two weeks from the day a test is returned to contest its scoring.

- Please be in your seat by the beginning of class. There is to be no food or drink in the classroom. Sleep is allowed only in the event of a stultifyingly boring lecture.

- Signing up for and keeping current with the *Programming Languages* Forum at [http://beastie.cs.ua.edu/forum](http://beastie.cs.ua.edu/forum) is mandatory. Schedule changes, assignment clarifications, and such will be announced via the forum. Registrations will not be allowed after that the first week of class. If you don’t register, you will be allowed to read the forum but you will not be able to post.

- Programming assignments are to be submitted electronically by 4:59:59 p.m. on the day they are due. For these projects, there is a grace period: late projects will be accepted up to 48 hours after the due date with no penalty. *Please do not ask for an extension as I will most assuredly say no and then we’ll both end up feeling bad.*

- Each assignment is to be done individually. There can be no sharing of code or solutions.

- The University of Alabama Code of Academic Conduct will be rigorously enforced. Please review the code at [http://www.studenthandbook.ua.edu/academicintegritypolicies.html](http://www.studenthandbook.ua.edu/academicintegritypolicies.html) and be sure to follow this Code in all activities associated with CS403/CS503. Any violations of this code will be handled as an Academic Misconduct case. The minimum punishment requested for plagiarism, or any other academic misconduct, is dismissal from the class with a failing grade. You are required to protect your work from plagiarism. If your work is plagiarized, it will be assumed that you were a willing participant in the plagiarism and you will receive the same punishment as the plagiarist, absent extraordinary evidence to the contrary.