Syllabus for CSC 453: Compilers and Systems Software

Course Description

This course covers Compilers and related Systems Software. What is a compiler? What does it take to write a compiler? What are the tools used to write a compiler? How are the lessons of compilers applicable to systems software?

This class centers on one large programming project (a compiler) that will be broken up into 5-6 programming assignments. The final product will be a working compiler for a small language. You will implement all the pieces of the compiler by yourself. Repeat: by yourself.

The project will be written in C, using many systems tools such as gcc, make, valgrind, flex, bison, debuggers (ddd, gdb) and standard GNU tools like gcc. These tools should all be familiar.

One minor goal of this class is to become intimately familiar with C: the only way to do this is to write a large piece of software in C and immerse yourself in the code.

Prerequisites

These prerequisites are non-negotiable. If you have not already completed the classes below, you will be administratively dropped from the class.

cs352: You need to know how to write in C, use Makefiles, use valgrind, write UNIX scripts for testing, and in general feel comfortable with C on UNIX.

cs252: You will need to know how to write assembly language.

cs345: You will to feel comfortable with theoretical formalisms, discrete math, data structures (trees especially).

cs372: (optional) This class gives you a lot of background which can be very useful.

This class is hard: you will need the prerequisites under your belt in order to make progress. This class is a combination of theory and practical programming, and you need both to survive.

Staffing

Lecturer: Richard T. Saunders

Email: saunders@cs.arizona.edu  Office: Gould Simpson 722

Phone: You can call the front office at 621-4632, but email is the best way to get a hold of me or the discussion section leader.

Lecture Class Times: Monday, Wednesday 10:00am - 11:15am

in Gould-Simpson 906

Office Hours: Monday, Wednesdays 4:00-5:15 in office

This may change during the semester, but frequently, I will stay in GS 906 after class answering many questions after class before returning to my office.

Materials

Compilers: Principles, Techniques and Tools (Second Edition)

Required: This is the "purple dragon book" (not to be confused with the red dragon or the green dragon book). This book is considered (historically) the standard compilers book. We will frequently refer to the book throughout the semester for other examples, exercises, and for structurally how we'll proceed. Nominally, we plan to cover Chapters 1-8. Later chapters may be useful for you outside of class.

**Flex and Bison**

Optional: This is an O'Reilly book. It's a very practical book for using Flex and Bison (both tools we will use in class). A lot of material on Flex and Bison is available on-line, so this isn't required, but it may be useful to have another perspective.

by Harbison and Steele, ISBN 0-13-089592

Optional (but highly recommended): This is the C book. Whenever you have any questions about C or an implementation, this is the book. We will be referring to it throughout the semester and if you use C in your next job, it will be a highly prized commodity.

**Discussion Section**

Make sure you enroll in both Lecture and Discussion Section! It meets Friday 9:00-9:50am in Gould-Simpson 906.

The discussion section is run by the TA: this section will be for discussing details of the programming assignment. The TA, email, and office hours:

Saurabh Maniktala <saurabhmaniktala@cs.arizona.edu>
Tuesday 11am-12Noon, Friday 10am-11am In Gould-Simpson 710 E

Saurabh will be also monitoring the pizza group to help with questions and clarifications.

**Class Web Presence**

**Web Site:** [http://cs.arizona.edu/classes/cs453/fall13](http://cs.arizona.edu/classes/cs453/fall13)

The web site will contain the syllabus, assignments, some handouts and important announcements. If there are some major problems with the assignment, clarifications and extensions will be announced on the web site! Check early, check often.

We will also use piazza.com for discussion and clarifications on assignments. Be courteous! Any malevolent behavior will shutdown piazza.

All assignment points will be posted on D2L, as well as turnins will happen on D2L.

The final grades will NOT be posted in D2L, only on the final UAccess web site. (So don't email me about this)

**Attendance and Lecture**

Attendance is not recorded. If you miss a day, you will not be able to make-up the exercise and/or test.

*You are, in general, responsible for everything that happens in lecture*
Accounts and Labs

Labs To Work In

There are two labs in Gould-Simpson that you will have access to:

<table>
<thead>
<tr>
<th>Lab Name</th>
<th>Floor Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gould Simpson 228</td>
<td>2nd floor</td>
</tr>
<tr>
<td>Gould Simpson 930</td>
<td>9th floor</td>
</tr>
</tbody>
</table>

Your CATCARD should get you into either lab 24 hours a day. You should also have 24-hour access to Gould-Simpson building via the door just south of the 228 lab. (Note, the card reader is ONLY on the south door of Gould-Simpson)

Just swipe your CATCARD into the lab card reader or the door reader.

By registering for the class, your CATCARD should be programmed to open the labs and south Gould-Simpson door. If you registered late or have problems, email lab@cs.arizona.edu.

Accounts

The purpose of this class is to learn how to build a compiler: to do that, you will have to be using the computers in the Gould Simpson labs (see above). You can use your own computers to develop and program, but everything must work in the Gould Simpson computers: that's where the assignments are turned in and graded. (We will run all testing on lectura.cs.arizona.edu).

Computer accounts on CS department machines will be created automatically for you. The account name will be the same as your UANetId. You will be sent an email that your account has been created. The email will contain the initial password for the account. When you first logon, you will be prompted to change your password. The lab sections this week will provide you an opportunity to logon to your account and change your password. You will need your initial password when you come to the section meeting. If you do not yet have a UANetId and/or an email.arizona.edu email account, you need to create these. Go to https://account.arizona.edu/ to create them. Both are needed in order to get CS computer accounts.

The section leaders can help you with this in the first two labs if you can't log in directly.

Grading

The grades are based on 3 components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage of Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Assignments</td>
<td>60% of your grade</td>
</tr>
<tr>
<td>Midterm</td>
<td>20% of your grade</td>
</tr>
<tr>
<td>Final</td>
<td>20% of your grade</td>
</tr>
</tbody>
</table>

Grades will be assigned on the standard scale:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90% - 100%</td>
</tr>
<tr>
<td>B</td>
<td>80% - 89%</td>
</tr>
<tr>
<td>C</td>
<td>70% - 79%</td>
</tr>
<tr>
<td>D</td>
<td>60% - 69%</td>
</tr>
<tr>
<td>E</td>
<td>0% - 59%</td>
</tr>
</tbody>
</table>

You will get at least the grade corresponding to your percentage. There may some rounding up on edge cases: usually, we will tend to bump you up if you do better on later assignments (when they are harder).
Programming Assignments

You will submit your assignments using D2L. Instructions on how to do this will be provided with each assignment.

You are encouraged to complete assignments on time.

SPECIAL NOTE: Unlike other classes, the next assignment depends on you getting the previous assignment correct. The Lexer you create in assignment 1 will be used to build the Parser in assignment 2 which will be used to create the AST in assignment 3 and so on. DO NOT FALL BEHIND or you may not be able to catch up.

Late Days

We do accept late programming assignments, under the following rules: You can use one late day on a single assignment. Each student may use up to a total of three (3) free late days for programming/homework assignments. A “late day” is defined as one minute up to 24 hours after the due date/time. Weekends and holidays count as late days. Once you have exhausted your three late days, you may still submit a late assignment. Such assignments will be penalized 20% for being one day late.

Once solutions have been provided in class, WE CANNOT ACCEPT ANY LATE ASSIGNMENTS.

Collaboration & Cheating

You need to do the work yourself: it's the only way to learn the material. It's okay to ask other people general concepts:

• “How does a loop work again?”
• "What is the syntax of SPIM again?"
• "Where would I find documentation and examples on the web or the book?"

but you must write your own programs:

If you are copying someone else's code, you are cheating.

If someone else is writing a program (or a piece of a program) for you, you are BOTH cheating.

If you feel like you are lost and want to cheat, PLEASE DON'T! Myself and the TA are always available for help, for a different perspective, or a different personality. The book has a good index, the web has lots of examples to look at. There is a TA and myself who want to help you: please contact us if you are struggling.

The general rule is Do the work yourself

Violations

Students who violate the Code should expect a penalty that is greater than the value of the work in question up to and including failing the course. A record of the incident will be sent to the Dean of Students office. If you have been involved in other Code violations, the Dean of Students may impose additional sanctions.

The CS department policy can be found at:

http://www.cs.arizona.edu/policies/collaboration.html
The UA’s “Code of Academic Integrity” can be found at:

http://deanofstudents.arizona.edu/codeofacademicintegrity

Both provide further details on what constitutes cheating, the penalties that may result, and the procedures involved.

Protect Yourself

- Do not allow others to know your password (a good policy in general for all passwords).
- Do not allow others to use your computer account.
- Do not leave printed copies of your program in the lab (the labs have secure white paper recycling).
- Do not allow others to use your personal computer without your supervision. * Protect your homework solutions in all such cases.
- Always logout of lab computers when you leave the lab.