CS 321 Languages and Compiler Design I - Winter 2012

Instructor:
Andrew Tolmach
120-23 FAB
503-725-5492
email: `apt@cs.pdx.edu`
Office Hours: M1-2 Th4-5 & by appt.

Course home page: `http://www.cs.pdx.edu/~apt/cs321`

Description

CS321/322 studies, in parallel, the design and implementation of programming languages and the design and implementation of compilers. The course is centered around a substantial programming project: implementing a complete compiler for a realistic language. We will study formal methods for specifying the syntax and semantics of languages, and, in some cases, use tools based on these methods to help build the compiler. We will examine the structure and capabilities of programming languages with an emphasis on the demands of efficient implementation. A detailed list of lecture topics may be found below.

Prerequisites

Formal prerequisites are CS201, 202, 300, and 311 with grades of C or better. Substantial experience with the C, C++, C# or Java programming language is *essential*. Java will be used as the implementation language for the compiler project, but previous experience with Java programming is not required. Experience using program development tools (compilers, linkers, debuggers, etc.) and some prior exposure to basic automata theory will be very helpful. For CS322, some experience with machine-language programming will be extremely desirable.

Texts

The required textbook is

• Keith D. Cooper and Linda Torczon, *Engineering a Compiler*, Morgan Kaufmann. Second edition (2011) is preferred; first edition (2004) is acceptable.

Additional required readings from other sources may be handed out in class or made available on the web page.

You'll also definitely want access to a Java textbook. There are many good possibilities; here are two I suggest:

• Ken Arnold, James Gosling, and David Holmes, *The Java Programming Language*, 4th ed., Addison–Wesley, 2005. (Earlier editions don't cover all the features of Java 1.5/6.)

• Bruce Eckel, *Thinking in Java, 4th ed.*, Prentice-Hall, 2006. (Can also be purchased in an electronic edition. The third edition of this book is available free on the web at `http://www.mindviewinc.com/Books/downloads.html`, but again, it only describes Java 1.4.)

You will also need documentation for the particular version of Java that you choose to use (typically available online) and for two Java-based tools that will be needed in the project:

• JFlex (`http://jflex.de/`)

• CUP (`http://www2.cs.tum.edu/projects/cup/`)

Lecture notes will be available electronically in pdf format via the course home page, either before or shortly after the lecture occurs. However, lectures may not follow the notes precisely; it is up to you to take additional notes during class.

Project

Over the course of CS321/322 you will write a complete compiler for the **fab** programming language. A description of the language will be made available on the course web page. Your compiler must be written in Java.

Exams

There will be one mid-term and a final exam. Both are *closed-book*. Exams will cover topics from lectures and readings, emphasizing material that is not directly relevant to the programming project. Not all the material is covered in the readings, so lecture attendance is important. Exams are scheduled in advance; unless prior arrangements are made, a grade of zero will be recorded for missed exams.

Homework problems may be assigned from time to time as an aid to help you study for the exams; they will not be collected or graded.

Grading

Approximately 1/2 on programming assignments and 1/4 on each of the two exams (including the final).

Programming assignment grades will be determined primarily by observing program behavior on test inputs. Computer programming being what it is, this policy means that "small" errors in your code can have a large effect on your grade. Sample test inputs and a correctly-behaving executable will be provided for you to compare your program against. It is your responsibility to apply these tests *and others of your own devising* before submitting your assignment. We will apply some non-public tests to your programs as well.

Computing Facilities

All parts of the project will be implemented in Java. For CS321, you may develop your project solutions on any machine and operating system you like, so long as it supports Java (JDK version 1.6 or later). However, if you want assistance from the course staff, you should do your development on unix or linux; help for Windows will be limited.

You may choose to use the PSU CS machines to develop your project solutions. If you don't already have an account, go to FAB 82-01 to obtain one. Java 1.6 is available in the default environments of the CS linuxlab machines, and can be obtained on the Solaris machines by using addpkg of java_6. Documentation for these versions of Java can be found on the web at http://download.oracle.com/javase. Files associated with assignments will be made available for download via the course web page.

Submitting Programs

Programs should be submitted electronically; details on where and how to submit programs will be provided with each assignment.

Mailing List

Important information will be distributed throughout the term via a mailing list called cs321list. To subscribe to the list, visit https://mailhost.cecs.pdx.edu/mailman/listinfo/cs321list and follow the directions there. Please mail questions to the instructor directly (at apt) rather than to this list; the instructor will copy mail of general interest to the list. The list is archived so you can consult previous messages.

Tentative Schedule (subject to change)

Date		Asgn	Reading	Topics
Jan	10	0	CT1	Introduction
	12		skim Java	Reviewing Java
	17	1		Language Processing in Java
	19			Language design; Historical survey
	24	2	CT2	Lexical Analysis; Regular expressions
	26		JFlex	Finite automata; Scanner generation
	31	1*	CT3.1–2	Syntactic analysis; Grammars
Feb	2		CT3.3	Top-down parsing; Recursive descent
	7			Table-driven LL parsing
	9	2* 3	CT3.4.1	Bottom-up parsing
	14			**Midterm Exam** (in class)
	16		CT3.5,CUP	Parser generators
	21		CT4.1,4.3–4	Syntax-directed translation
	23		CT4.2	Data types
	28	4		Type declarations and equivalence
Mar	1	3*		Type checking
	6			Type Polymorphism
	8			Abstract data types and modules
	13			Object-oriented programming concepts
	15	4*		Review
Mar	19			**Final Exam** (**Monday** starting at **10:15 am**)

Readings column key: CT = Cooper and Torczon textbook, **second edition**; 2nd ed. 3.4.1 = 1st ed. 3.4; 2nd. ed. 3.5 = 1st ed. 3.6. Java = your preferred Java textbook; JFlex = JFlex user manual; CUP = CUP user manual.

Additional readings will be assigned from time to time. Assigned selections should be read *before* the associated date.

Assignments

Programming assignments, listed below, are distributed on the dates the assignment number appears in the schedule above and are due when the starred number appears.

- 0   Java warm-up (not to be turned in)
- 1   simple language processor
- 2   **fab** lexical analysis
- 3   **fab** parsing
- 4   **fab** type checking

Rules

Programs are due *by 1:30 p.m.* on the specified due date, i.e., just *before* class. Late programs are *not* accepted except in extraordinary circumstances, and then preferably by prior arrangement. The project deadlines are there to help you, by forcing you to keep up during the term.

All programming assignments must represent your own, individual work. It is permissible to discuss the assignment with other students, but you must develop the solution yourself (although you can consult the tutors for help in debugging). *Do not, under any circumstances, copy another person's program and submit it as your own.* Writing code for use by another or using another's code in any form (even with their permission) will be considered cheating. Cheating on an assignment or exam will result in an automatic zero grade for that piece of work, and the initiation of disciplinary action at the University level.

If you are a student with a disability in need of academic accommodations, you should register with Disability Services for Students and notify the instructor immediately to arrange for support services.