Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.035, Fall 2000 Handout 3 – General Information Wednesday, September 6

Course Title

Computer Language Engineering

Prerequisite

6.170 and
proficiency in Java

Texts

Engineering a Compiler
Keith Cooper and Linda Torczon
This book is not yet published; we are a beta site.
Bring the voucher handout to the cashier's office (10-180)
along with $14.00.
Take the receipt to 38-501 to pick up a copy of the text.

Your help is needed to debug the book;
please send us any comments you may have!
A few chapters will be missing initially;
more will become available as time goes on.

Optional Texts

Modern Compiler Implementation in Java (Tiger book)
Andrew W. Appel
Cambridge University Press, 1998

Advanced Compiler Design and Implementation (Whale book)
Steven Muchnick
Morgan Kaufman Publishers, 1997

Compilers – Principles, Techniques and Tools (Dragon book)
Aho, Sethi and Ullman
Addison-Wesley, 1988

Lecturers

Saman Amarasinghe
NE43-620B, x3-8879
saman@lcs.mit.edu
http://www.cag.lcs.mit.edu/~saman/

Martin Rinard
NE43-620A, x8-6922
rinard@lcs.mit.edu
http://www.cag.lcs.mit.edu/~rinard/

Course Secretary

Rachel Allen
NE43-620, x3-9620
rachela@lcs.mit.edu
(Contact person for missed handouts, etc.
Please call or e-mail first to check that she is in.)

Course Home Page **http://web.mit.edu/6.035/www/**

Teaching Assistants	Nathan Williams NE43-803, x3-3447 **nathanw@mit.edu**	Matt Deeds NE43-618, x3-6284 **mdeeds@mit.edu**
	David Maze **dmaze@mit.edu**	Kenneth Lu x5-9741 (home) **kenlu@mit.edu**

Class Format

The lectures, problem sets, and exams of 6.035 are structured around the compiler project, which is divided into six segments. Each segment addresses a separate aspect of compiler construction. Each segment (except the first) consists of a series of lectures, a programming assignment, and a Problem Set or an in-class Quiz.

At the beginning of a segment, the programming assignment will be given. Next, a series of lectures will introduce the material. Each segment will have either a Quiz or a Problem Set. The Quizzes will be in-class (50-minutes). The Problem Sets will be given out with the programming assignment and will be due few days after the lectures series ends. Each segment will have a recitation on the project implementation details that will take place after the lectures (and the quiz). The remaining time of the segment will be devoted to the project; thus, no classes will be held. During this period, each group should schedule a “project review” meeting with your TA to discuss the design decisions, project progress, and group status. A segment comes to an end on the due date of the project assignment. Note that the first segment is an exception to many of these rules. Handout 1, "At-a-glance", outlines the schedule, and Handout 4, “Project Overview”, expands on the project details.

The Project

The main course work of 6.035 is the design and implementation of a compiler. In each segment of the course, a part of the compiler will be constructed. The six segments are the scanner, the parser, semantic checking, code generation, data-flow optimization, and instruction optimization. The first assignment (Scanner) is an individual project. The other six assignments are group projects.

As in previous years, we are offering two versions of this compiler project: an 18-unit version and a 12-unit version. All members of a project group should register for the same number of units. All students should register for 6.035 (12 units). Those wishing to do the more ambitious project should also register for 6.907 (6 units) by Add Date (Friday, October 6th). A professor’s signature is not required on an Add form for 6.907. The grade for the six extra units will be based solely on the project.

The 12-unit version of the project is for the health-conscious: You will be served *Decaf* to get a hands-on experience of building a simple yet complete compiler. In the 18-unit version of the project, you will be served a strong, steaming *Espresso* to enable you to also experience some of the real world problems that arise in building industrial-strength compilers.

We offer the more difficult project and the six extra units to provide interested students with an opportunity to gain a deeper understanding of the issues involved in building a compiler. This should be seen as the reason for undertaking the 18-unit version of the course, not the desire to simply get six extra units.

The due dates will be the same for both versions of the project.

The first assignment (Scanner) is the same for both Decaf and Espresso.

Class meetings

All class meetings – lectures, recitations, and in-class quizzes – are scheduled from 11:05 a.m. to 11:55 a.m., Monday through Thursday. Lectures will be held in 3-370 on Mondays and Wednesdays and in 3-270 on Tuesdays and Thursdays. A tentative schedule is provided in Handout 1. Note that the lecture and recitation schedule is irregular.

TA Office Hours

Each TA will set his own office hours and will spend those hours each week in his office. These hours will be announced later. If you wish to see your TA at some other time, please call or e-mail him to arrange a meeting.

Section Assignments

After groups have been set up for the project, we will make recitation assignments. The recitation assignments will be posted on the web by end of Monday, September 11th. All members of a group will have the same recitation instructor.

Requirements and Grading

- The problem sets are due either at NE43-620 before 11:00AM or at the beginning of the recitation. We will be handing out the solutions in-class during the recitation. Thus, no late problem sets will be accepted.
- No make-up quizzes will be given. If you are unable to take a quiz for a legitimate reason, talk to one of the instructors BEFORE the quiz.
- Hand in the project write-up at NE43-620; it is due before 5:00PM. For late project assignments, a penalty of 10% will be imposed. Note, however, that the project has to be completed before the due date of the next project.
- For each project assignment, all members of a group will get the same grade. However, if we feel that a particular member of a group is contributing above and beyond his or her share or is not carrying the weight we will adjust his or her grade accordingly. Therefore, it is important that you keep your TA updated with the progress of your group and your contributions to the project.

The overall grade for the class will be assigned using the following weights:

For 6.035 (12 units)

Compiler project	54%	
Problems sets	16%	(8% each)
Quizzes	30%	(10% each)

For 6.907 (additional 6 units)

Compiler project	100%

Incompletes

Incompletes are only given in unusual circumstances, and only when work has been completed satisfactorily up to the point when the incomplete was requested. To quote the MIT Course Catalog: "The grade of *I* is to be used only for subjects for which a minor portion of the work required has not been completed, and when a passing grade is expected when completed."

Policy on Collaborative Work

The "Department Guidelines Relating to Academic Honesty" require that we inform you of our expectations regarding academic conduct.

For homework assignments and the first project assignment, you are permitted to discuss the problem requirements and background material with anyone. However, the actual solutions you hand in should be your own individual work. For rest of the project assignments, you may discuss background material, problem requirements, approaches to problems, and design with anyone, but you may not view any code written for 6.035 by anyone, including past students. The actual coding should be the work of your group only. All exams are to be done individually.

In the past, many academic misconduct cases have come about because of poor judgment on the part of students who find themselves in the position of starting an assignment late or feel that they are incapable of completing an assignment. Please bear in mind that the long-term consequences of an academic misconduct case will do much more damage to your career than the worst possible grade you can get in 6.035. Please think through the issues now so that don't lose perspective if you find yourself temporarily tired, stressed, desperate, and tempted to violate the policy on collaborative work during the semester.

Dealing with academic misconduct cases requires an enormous amount of nonproductive effort for the course staff. We will not be happy if we have to deal with one of these cases this semester, and we will prosecute to the fullest extent provided for by the University relations.