CSCI0320 Student Missive

Introduction to Software Engineering
CSCI0320, Spring 2009
Steven P. Reiss

1 Purpose

The purpose of this course is to provide students with the background they need to design and implement moderate (10-100k loc) software systems. The emphasis of the course is on learning the design and coding techniques that are needed in the implementation of such systems. Design techniques include understanding design from the class level through to software architectures, understanding how to express designs, understanding how to design user interfaces as part of such systems, understanding how to use threads and sockets, understanding file I/O and databases, and understanding how to integrate applications with the web. Coding techniques include proper use of tools, an emphasis on testing, memory management, systems programming, proper use of programming languages, and coding and programming styles. The course also serves as a basic introduction to software engineering, with students doing a team project from requirements through implementation. Here emphasis is on learning how to design for and work as a team, doing presentations, and that the primary goal when writing code is to enable it to be maintained.

2 Who Should be Taking the Course

Students who are interested in software design and engineering principles should take this course as an introduction. It is excellent preparation for a summer internship or future computer-related job, as it emphasizes skills required to work well in a team. Students outside the department take this course to develop the skills necessary for participation in a group project involving a large amount of programming.

We expect the students taking this class to have a strong sense of OO programming, working knowledge of UNIX, good understanding of data structures, and to be highly motivated. If you have taken CS 15/16, 17/18, or 19, you should be sufficiently prepared for this course (CS22 and CS31 will help as well). If you have not taken the prerequisites, you should talk to a TA to determine if the course is right for you.

Last year we combined the old CS32 and the old CS36 into a single course. The new course is roughly 2/3 software engineering and 1/3 systems programming. Students who have taken CS32 in the past and want to get the systems programming knowledge should sit in on the lectures from 3/11 through 4/24 and do the two associated assignments. Students who have taken CS36 in the past can still take this course. Please contact Professor Reiss for more information.

3 Content

The course basically covers four topics: advanced programming concepts, object-oriented (OO) design, software engineering, and systems programming. These components will be blended throughout the course, with the emphasis throughout being on understanding program and system design at all levels. The beginning of the course will emphasize programming concepts and small-scale design issues. The middle of the course will cover advanced programming using threads and sockets as well as system design concepts. The latter part of the course will cover systems programming issues, tools, and concepts.

4 Lecture, Labs, Presentations, and Sections

In this course we will use a variety of different approaches to teaching design and programming. We will have traditional lectures, given by Professor Reiss. These will be focused on design issues and should provide the basic principles and techniques you will need to master in order to design the latter programming assignments and especially your final project.

Lectures, however, aren't the best way of teaching hands-on material. Programming techniques and concepts are best learned through doing. Rather than having massive programming assignments that cover all the topics that are necessary, we will have weekly lab sessions (you only need to go to one). These two hour sessions will be run like a typical science lab: there will be a handout describing what you should do and what material you should read in preparation for the lab, then you will be expected to complete a simple programming or related assignment during the lab time under the supervision of the TAs.

There will also be two short homework assignments in this course. The purpose of the homeworks is to help students with the transition from the software engineering part of the course to the systems programming topics.

An essential part of doing a large project and of software engineering in general involves doing presentations, telling others what you propose to do, what you are doing, and what you have done. We will provide you with the opportunity to make such presentations throughout the course. We will have an evening of specification presentations where you will have the opportunity to describe what you plan to do. We will have another evening of design presentations where you can describe how your system will work. We will have several labs dedicated to letting you present an early version of your project and get praise and feedback on it. Finally you will have the opportunity to demonstrate your project for the class and the rest of the department.

The TAs will be holding a help session for each assignment. Additional sections or mini-help sessions will be arranged if and when they are needed.

5 Textbooks

The textbooks are available in the bookstore.

Vermeulen, et al, *The Elements of Java Style*

This text, which is required, provides a set of style and coding guidelines that is a good starting point for serious Java programming. (We will provide our own additions and suggestions in handouts and on the web as the course progresses.) While this book is restricted to Java, most of the principles apply to C and C++ programming as well.

Whittaker, *How to Break Software*

This text, which is optional, provides a set of guidelines for testing software, which is a topic we will stress throughout the course. Thinking about testing as done in the book helps you produce more robust software with fewer errors that requires less debugging time.

Braude, *Software Design*

This text, again optional, provides a nice compact overview of several topics that we will cover in the course in parts I and II. These include a nice overview of software engineering, various design issues, UML, and design patterns.

6 TAs

Head TAs:	Jacob Frank (jfrank)
	Max Rusnak (mrusnak)
UTAs:	Ben Herila (bherila)
	Travis Webb (jtwebb)
	Lyn Fong (lfong)
	Ben Martada (pmartada)
	Ryan Zelen (rzelen)
	Jon Silverman (jds)
	Josh Dawidowicz (jdawidow)

TA hours will be held in the fishbowl/birdcage on the 2nd Floor; shopping period hours are posted on the website.

7 Required Coursework

This course is comprised of labs, homeworks, programs with graded designs, and a final project.

The labs are designed to teach important course concepts. Attendance is mandatory; they will be graded by a TA in class based on completion of each subsection of a lab assignment. Labs are designed to be successfully completed in the given lab time; if more time is necessary, a student is encouraged to attend another lab section or complete the lab on their own time get it checked off during TA hours.

Lab times (subject to change):	
Sunday	6-8 pm
Tuesday	8-10 pm
Wednesday	6-8 pm
Thursday	4-6 pm
Thursday	6-8 pm

All labs are held in the Sunlab on the first floor of the CIT.

The programs will range from smaller individual projects, meant to familiarize you with advanced Java programming and object-oriented design techniques, to larger collaborative projects. They will cover advanced programming, use of available programming tools, and user interface and web programming, and will culminate in a substantial group project at the end of the semester.

In order to pass CSCI0320, you need to have a passing final grade in the course, and you also need to hand in a working version of all assignments. Failure to handin a working version of all programming assignments (regardless of your final average) will result in an automatic NC. The last day to hand in a working version of an assignment is Friday, May 9 at 5pm. See Unacceptable handins below for more information.

Programming assignments are not merely graded on their functionality: there will be a special emphasis on good software engineering practices. This means that we will expect you to produce readable, well-documented code, to use modularity and data protection, and to spend time thinking about the design and breakdown of your program before you start coding. The TAs will read your code. A perfectly working program which is not well-designed, well-documented or readable will not receive an A.

Extra Credit: There will be ample room for bells, whistles, and other credit-garnering efforts on the part of ambitious programmers. You are invited to get creative, as long as it does not make you late. Remember to always implement the program according to assignment specifications before launching off into extra credit work. Rewarding bells and whistles with extra credit is left to the discretion of the TAs (and Professor), so you may want to discuss ambitious plans with a TA to make sure they are considered appropriate for credit. In general, efforts outside the intent and focus of the assignments will receive small amounts of credit.

Handing In: Your working code should be handed in online by the deadline announced in the assignment's handout; the TA staff will announce the exact handin procedure on each handout. You must hand in your assignment from your account.

Please keep in mind that TAs have a substantial workload, and that grading stacks of assignments is not the most exciting thing to do. If you make a small effort to present your work in an organized fashion, then that will make life considerably easier for the TAs.

Unacceptable handins: Occasionally, a student will hand in a program that is so far from meeting the requirements of the assignment that the grading TA will consider it unacceptable. In that event, the assignment will be registered as not done. In order to pass CSCI0320, you need a passing final average and you need to have handed in a working/acceptable version of all the assignments. Even if you have a passing average, you will still receive an NC for the course if you do not hand in an acceptable version of all the assignments by Friday, May 9 at 5pm - three days after final project presentations. This deadline is set by the university and will be strictly adhered to without exception. Since it is extremely hard to state explicitly what is and is not considered acceptable, it is the student's responsibility, when his or her work is not at all close to meeting the assignment's specifications, to check whether the work meets minimal requirements by conferring with the Head TAs. You can keep in mind that any program with no functionality or any program with poor coding style (i.e. no comments and in severe violation of the CSCI0320 coding standards) will automatically be considered 'unacceptable.'

8 Grade Breakdown

The course grade will be based on the in-class projects, programming assignments, class participation and the final project. The following percentage breakdown will be used:

Program 1 (Solar)	10%
Program 2 (Galaxy)	10%
Program 3 (NewsWhere)	10%
Program 4 (FileView)	10%
Program 5 (Chatter)	10%
Final Project	25%
Labs	15%

Homeworks	5%
Class Participation	5%

Designs will be counted as a percentage of that project's grade.

If you disagree with any grade you receive, please first see the TA that graded your assignment.

Group Grading Policy: The TAs far prefer that each group has a single electronic handin per program. The grade assigned will be a group grade, i.e. the same for all members. However, occasionally the situation arises where one team member feels that another member's contribution is substantially deficient. In such cases, the team members should first make a serious effort to deal with this problem and to hand in a working program together; indeed, part of working in a group is resolving differences between group members, and motivating your teammate(s) to work in an expeditious manner. Failing this, the person whose work is completed may submit a separate handin. The README should indicate that you are handing in separately, and the Head TA should be notified via email. Next, this person must meet with the TA who is grading the program on hours, and demonstrate that his or her portion of the program works. This will probably necessitate writing some test code. Finally, at some point the group should hand in the integrated code for the program. This should be handed in by the other person in the group to avoid overwriting the first partial handin.

If you believe that you are in this situation, you are strongly encouraged to discuss it beforehand with a TA. In addition, the TAs reserve the right not to accept separate handins from a group if any of the above conditions are not met.

9 Deadlines and Late Handins

Assignment deadlines are intentionally spread out because the course is intensive. You will find that if you start each assignment when it is handed out (rather than near when it is due), the workload will be manageable. However, if you are late with one assignment and that causes you to compress the work on the next, you will probably find that the course load will quickly become unbearable. Moreover, the latter assignments are team efforts, and to be fair to your other team members, you must not get behind in the course material.

Deadlines: Tentative due dates for the assignments are listed in the syllabus, but always note the actual due date on an assignment handout when it goes out and note any changes announced in class.

Late Days: We will give you two "late days" to use at your discretion over the course of the semester. Each provides you with a 24 hour extension on a programming assignment. If you use late days on a group project, they apply only to you and not to the group as a whole.

Late Handins: Your TAs realize that sometimes you may have scheduling conflicts such as an exam on the day your program is due. Unfortunately, we can't give extensions for academic overload; it is your responsibility to plan carefully. However, you do have the option of handing your program in late; your grade will then be subject to penalty, depending on the lateness of your handin. A percentage of the score you receive will be deducted as follows:

HANDIN BY	PENALTY
1 day after due date	10%
2 days after due date	20%
3 days after due data	30%

4 days after due date	40%
later than 4 days	100%

There are several remarks on the late policy. First, this policy does not apply to the final project, homeworks, or for designs, for which there are no late handin possibilities. Second, days on which the Sun Lab is not open (such as over spring break) will not count against you. Third, late penalties have nothing to do with the determination of an assignment as working/acceptable or unacceptable. And finally, please note that handins later than four days after the original deadline will not receive any credit, and will thus endanger your chances of passing the course. It has been shown in the past that CSCI0320 is simply too tight a course for people to be able to fall behind and still do well. While this late policy may seem harsh, it is designed to make you recognize the importance of meeting your deadlines.

There is no extra credit for handing in early.

Extensions: As a general rule, no extensions will be granted in CSCI0320. Academic overload is *not* a valid excuse for an extension. Extensions *may* be granted by the professor only for medical or other reasons which are deemed acceptable. In the case of medical excuses, a note from Health Services is required; in other cases, a note from a dean would be appropriate. Of course, extensions for individual members on team projects are a more complicated issue, and these will hardly ever be given. To request an extension, mail `cs032headtas`.

Incompletes: Incompletes for the course will be granted only in extreme circumstances. You should be aware of the university policy that says that students can only be granted an Incomplete if their work up until the date of the Incomplete is satisfactory. If you need to file for an incomplete you should talk to Professor Reiss.

10 Accounts

All students will fill out a registration form on the first day of class, so we have your name recorded, and whether or not you require a new CS account.

11 Staying Up to Date

The primary mechanisms for communication of administrative information will be the website and mailing list. All students are responsible for reading either the website or the mailing list regularly – important information will be posted both places. The site FAQ will be used for student discussion and clarifications, so if you are confused about something, check the FAQ archives, and if you have a question, post to the FAQ.

12 Getting Your Questions Answered

There are many ways to get questions answered or to get problems solved. You can check out the CSCI0320 home page, talk to a TA during office hours, or post to the CSCI0320 FAQ. You *c*annot ask questions of individual TAs when they are not on hours; these must be sent via email (cs032tas@cs.brown.edu). This includes calling the TAs at home, writing the TAs, or mailing the TAs with questions that should be asked on hours. The TAs strongly encourage you to be resourceful. They expect that you have made every effort to figure out a question or problem, using the many resources you have available to you, before you go to them. If you put a concerted effort into solving the problem yourself (or with your partner for group programs), the TAs will be more than happy to assist you. If you have not, the TAs will likely refer you to documentation, and send you away to look up that information.

CSCI0320 Documentation Page: We have created a page off of the CSCI0320 home page which contains links to many useful resources and sources of documentation. Also, if you find any good sources of online docs, let us know, and we can add that link off the docs page.

JavaDoc: JavaDoc for all the standard Java packages is available at `http://java.sun.com/j2se/1.5.0/docs/api/`. JavaDoc locations for the various support libraries for the assignments will be listed in the assignment handouts or on the website.

CSCI0320 FAQ Page: If you have a question, problem, or something to say which you feel may be of interest to other students in the class, post it to the CSCI0320 FAQ. The TAs will regularly read and respond to these postings. Note that it is in your best interest to frequently read the postings in the CSCI0320 FAQ, and many will be pertinent to your assignments.

Questions about packages used for assignments can be asked in the FAQ *a*fter you make an attempt to find an answer in the documentation, either online or available on paper in the lab. Feel free to answer questions asked by other students or point them to helpful documentation you've found, but be careful not to share information that's not permitted by the collaboration policy. When in doubt, consult the TAs. You should not use the FAQ to share code, ideas, or algorithms, nor should you make non-course related postings to the FAQ.

TA Hours: As mentioned above, TA hours are posted on the website. Professor Reiss's hours are available by appointment; send an email to `spr`.

E-mail: If you have a personal concern or a question which isn't of interest to the class at large, you can mail the TAs about it by sending mail to `cs032tas@cs.brown.edu`; this goes to the entire course staff except Prof. Reiss. You can mail only the Head TAs and Prof. Reiss by sending mail to the `cs032headtas@cs.brown.edu`. Professor Reiss's email is `spr@cs.brown.edu`. In general, the TAs will read their mail daily and will respond as soon as their schedule allows. If you don't receive a response in a reasonable amount of time, please let us know; we receive a lot of emails and yours may have been lost in the shuffle.

13 Working from Home

We do not officially endorse working from home on course projects. If you decide to work from home, it is your responsibility to research and implement a way to do so. TAs will give priority to questions related to course material and in many cases individual TAs will be unable to help you figure out a way to work from home, but if you email the staff (cs032tas@cs.brown.edu), we may be able to help you. Most importantly, though, if you decide to work from home, make sure to *test out all your work in the sunlab before handing it in.*

14 Feedback

We're always interested in ways to make CSCI0320 a better course. You are urged to bring any suggestions and/or comments you may have regarding the course, the assignments, the lectures, the labs, the practicums, or anything else to the TA staff or Professor Reiss. All comments (both positive and negative) will be much appreciated.