

# Course Missive

*Fall 2009*

## Course Staff

<b>Professor</b>	Phone	Office	Email (@cs.brown.edu)
Prof. Pascal Van Hentenryck	863-7634	CIT 301	pvh

<b>Head Teaching Assistants</b>	@cs.brown.edu
Claudiu Saftoiu	csaftoiu
Jon Silverman	jds
Siddhartha “Sid” Jain	sj10

<b>Teaching Assistants</b>	@cs.brown.edu
Andrew Ayer	andrew
Benjamin Leib	bgleib
Donald Kendall	dkendall
Grant Kellerman	gkellerm
Kyle Cackett	kcackett
Kenneth Coane	kcoane
Liam Elberty	lelberty
Peter White	pjwhite
Shaun Verch	sverch

## Introduction

Welcome to CS 31 (officially known as CSCI0310), Introduction to Computer Systems.

This document provides you with a lot of important information about CS 31, and you **need** to read it to understand how this course works; you’ll thank yourself later, and the TAs will love you for it.

You may also want to refer to the list of lectures and assignments, which can be found on the course website.

Announcements made during the semester will be sent to the listserv and posted on the web page.

The course is intended primarily for sophomore computer science students and computer engineers. It fills a core requirement for the A.B. and Sc.B. tracks in computer science, and is required for the Computer Engineering concentration. This course is a prerequisite for CS 32.

### From the Course Announcement

Basic principles of computer organization. Begins with machine representation of data types and logic design, then explores architecture and operations of computer systems, including I/O, pipelining, memory hierarchies. Uses assembly language as an intermediate abstraction to study introductory operating system and compiler concepts.



### Prerequisites

In order to take CS 31, you should have taken CS 16 or CS 18 (or, with Pascal's permission, CS 4 or EN 3). CS 22 is also recommended.

However, it is most important that you:

- **Understand concepts of high-level programming.** You should be familiar with modularity, generality, and the importance of understandable comments. You should also understand arrays, recursion, and have a fairly good idea of how these concepts compare with other alternatives.
- **Understand object-oriented design.** You should be able to understand small system design and navigate class and method documentation.
- **Know Java well.** You should be able to program productively and understand object-oriented design and flow-of-control concepts. **If you don't already know Java, talk to the head TAs.**

### Lectures

Lectures will be given by Professor Van Hentenryck MWF at 2pm in the Motorola room (CIT 165). If the location changes, we will make an announcement at the preceding lecture and send a message to the listserv.

Note that the published syllabus is a plan, not a contract. It is (evil music here) subject to change without notice; *don't* rely on it to choose lectures to attend.

In addition to the regular weekly lectures there will be several lectures held in the evening throughout the semester. Their exact time and location will be announced in class. These lectures will address material which is relevant to topics discussed in class but will examine them in greater depth than normal lectures allow. Students will not be held accountable for material covered in these lectures, but are encouraged to attend.

## Assignments

Assignments for this class consist of 8 weekly homeworks, five projects, a midterm exam, and a final exam.

Homework is (in most cases) assigned each Wednesday after lecture and is due the following Wednesday at 2:00 pm, in the CS31 handin bin outside the birdcage. We will accept any homeworks handed in before we start grading them, but do this at your own risk!

Homeworks will be graded and returned to you within the week. Your worst homework will be counted at only 50%. The intent here is to make sure you do all the homeworks, which are essential to understanding the material, while still giving you a break on that one homework you didn't understand at all.

The first project requires you to implement a simple RISC with diglog, a circuit simulation program. The remainder of the projects require you to code: the second and third in MIPS assembly, and the fourth in Java.

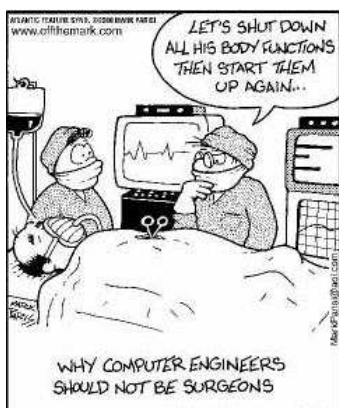
The first and third projects will have an interactively-graded design check-point worth 20% of your grade for the project.

<b>Project</b>	<b>Out</b>	<b>Design Check</b>	<b>Due</b>
RISC	October 2	Yes	October 16
Chars	October 16	No	October 23
Life	October 23	Yes	November 9
Maze	November 9	No	November 20
Compiler	November 20	No	December 9

The midterm on October 21 will be open-book and open-notes covering material through Moon-2. You will have three hours for this exam. The final, however, will be a take-home exam, date TBA.

## Grading

Your grade for the semester will be determined as follows:



Homeworks (8)	10
RISC	20
Char	5
Life	15
Maze	15
Compiler	15
Midterm	10
Final	10

Final grade cutoffs will be determined by the professor at the end of the semester. In general, questions about homework or project grades should be sent to [cs031headtas@cs.brown.edu](mailto:cs031headtas@cs.brown.edu), not the UTAs. If you have any other questions about your final grade, please get in touch with the professor, **not** the TAs.

Extra credit may be made available on course assignments. Points are totaled separately and will be added to your grade after final cutoffs for homeworks, projects, and exams are determined. The exception is Compiler, where the extra credit can be used for points on the final exam.

Late projects will be accepted with a 20% penalty per day late. You will have one late day, which you can use on any project, that will excuse you from this penalty for 24 hours. Projects handed in more than three days late will not be graded. However, **You must hand in a working version of all projects in order to pass the class.** A working version of a project executes successfully for at least one non-trivial test case. The last day we will accept handins of old assignments is December 8.

Project design checks must be done on the dates specified; there are no late design checks.

Homeworks handed in after we have begun grading them will not be accepted.

A Dean's excuse or a note from health services is required for any kind of exception to be made to the rules above. Once you have one of these you should talk with Professor Van Hentenryck, who is the only person who can give extensions.

## Collaboration Policy

In order to help the course staff evaluate each student in CS 31 as fairly and individually as possible, we have written a collaboration policy by which we expect all students to abide. **Please read this policy carefully**, as it may differ from collaboration policies in CS classes you have taken previously.

We strongly encourage teaching and learning with your peers. At the same time, your work must, in the end, represent your own understanding of the material. In that spirit, our overall policy is that you can talk about almost anything and work out solutions together, but that no notes may be taken away from collaboration sessions and your final write-up must be your own work (so-called "whiteboard collaboration").

## Permissions

You must ensure that your course documents have permissions which do not permit other students to view them. Ask a consultant or a TA for help with permissions if necessary, or use `man chmod`.

## Homeworks

You may work out solutions to problems together. However, you must understand everything you type up and hand in. To that end, we ask that you compose your solutions on your own and that you not take away any notes from any joint discussions (i.e. erase the whiteboard). When you are physically writing up your solutions, you should not be discussing the problems; your ability to reconstruct what you have developed with your classmates is evidence that you understand it.

You should not be looking at anyone else's write-up, nor should you permit

anyone to look at your write-up. Note that once you have started writing up your solution, you are permitted to continue to discuss problems, provided that your solution (and any old notes) are not used (i.e. you should be working off a clean sheet of paper or a clean whiteboard).

## Projects

Project design checks have the same collaboration guidelines as homeworks; given that you have done some thinking on your own, you may work through design ideas together. Your design check write-up must be your own, so you must separate before writing it up, and keep the write-up to yourself.

You may continue to discuss design after the design check, but should not share your write-up.

You may not look at or copy anyone else's code or circuit schematics for the purpose of helping you write your own, and you may not permit anyone to look at or copy your code (or circuit) for that purpose. You are permitted to work together on debugging, and can look at someone's code/schematic or work together to modify code/schematic for the purpose of helping him/her debug.

Testing ideas and a testing plan are part of project design, and you may collaborate to come up with them. Test code, however, must not be shared. Saying something like "Oh, have you tried doing an add after a jump? I had a strange problem with that." is fine, but "Try compiling this chunk of assembly code" or "Load this in to your program ROM" are not.

## Exams

Exams will be open book, but absolutely no collaboration is permitted. You may ask clarification questions of the professor and TAs during the exam, but may not communicate with any other student. Calculators, PDAs, and cell phones will not be permitted during the midterm. During the take-home final, you are expected not to discuss the exam at all with any other student. Questions on the final exam should be sent to the listserv `cs031@list.cs.brown.edu` (there will be no TA hours during finals).

## Course Materials

Copies of the lecture slides will be handed out at the beginning of each class. It is recommended that you take notes on and save these copies, as they will be a valuable resource in studying for exams and doing homework.

The textbook for this course is *The Essentials of Computer Organization and Architecture* by Linda Null and Julia Lobur (ISBN 076370444X), and is available at the Brown Bookstore, or your favorite online bookseller. **This text is not required or necessary.**

On the CS system, we will be using diglog (`/course/cs031/bin/diglog`) for circuit design and simulation, mipscope (`/contrib/bin/mipscope`) for simulation of a MIPS assembly machine.

The TAs will hold help sessions on both diglog and mipscope before you need to use them on assignments.

## TA Hours

TA hours will be held throughout the week in the Moon Lab, CIT 227, and the Fishbowl, CIT 271. Once the TAs work out their own class schedules, the exact hours will be posted on the course web page. TAs will gladly help clarify homework questions, explain concepts covered in homeworks, and help with general questions about course material.

TAs are here to help you, but remember, TAs are students too. **Please don't ask questions outside of official TA hours.** This includes talking to them in person or electronically while they are at home or in the lab.

If you feel that you can't possibly make the scheduled TA hours, please get in touch with the head TAs. If you need to speak with someone during business hours, try contacting Professor Van Hentenryck.

## Communication



You will need a **cs account** to complete the course assignments. When you turn in your signed collaboration policy, you will automatically be given a cs account if you do not already have one.

The **course web page** is an indispensable resource you will want to take advantage of. You can find online postings of assignments, solutions, lecture slides, announcements, TA hours, and other miscellanea. The course web page can be found at:

<http://www.cs.brown.edu/courses/cs031/>

Linked off the website, the **course listserv** ([cs031@list.cs.brown.edu](mailto:cs031@list.cs.brown.edu)) is your best means of contacting all the TAs and students regarding course material, assignments, and support code, and our primary means of contacting all students. A member of the TA staff should follow up on your email fairly quickly. Keep in mind that you are also responsible for any announcements sent to the listserv, so be sure to check your cs email regularly. You can also forward your email to another account by following the directions here: <http://www.cs.brown.edu/system/email/forwarding.html>

There are some guidelines by which you should abide when using the listserv.

- Before sending a question to the listserv, make sure you've tried to figure it out and consulted the lecture slides and handouts. Also make

sure you've looked at the previous listserv emails and checked to see if your question has already been answered.

- Do not send any inflammatory or offensive emails to the listserv. If you have opinions about the course, email them to `cs031headtas`.
- Do not send anything to the listserv that is specific to your program or your homework solution. These kinds of questions should be asked on hours or emailed to `cs031tas`. For example, do not email the listserv saying "this chunk of code does not seem to work, can anyone help me fix it?" and then attach the code.

If you have a personal question that won't be of interest to the class at large, you can **mail the TAs** at `cs031tas`. Please use good judgement and consider how urgent your problem is so that the TAs won't be swamped. Some of the TAs are assigned to monitor the mailing list and reply to any questions, so you should receive an answer promptly.

If you have administrative questions, comments about the course, or a problem with a TA, you should **mail the Head TAs** at `cs031headtas` or show up for their hours. Please use that address rather than mailing the Head TAs individually. You can also contact Prof. Van Hentenryck (`pvh`).