

# Course Missive

*Fall 2011*

## Course Staff

<b>Professor</b>	Phone	Office	Email (@cs.brown.edu)
Prof. Pascal Van Hentenryck	863-7634	CIT 301	pvh
<hr/>			
<b>Head Teaching Assistants</b>	@cs.brown.edu		
Andrew Ayer	andrew		
Benjamin Niedzielski	bniedzie		
Matthew Milano	matthew		
<hr/>			
<b>Teaching Assistants</b>	@cs.brown.edu		
Peter Kaufman	pk		
Michael Comella	mcomella		
Charles Yeh	ccyeh		
Sriram Jayakumar	sjayakum		
David Percy	dpercy		
Zach Davis	zdavis		
Ethan Langevin	ejlangev		
Sam Boger	sboger		
Jordan Place	jplace		
Max Gottesman	magottes		
James Sedgwick	jsedgwic		
Beatrice Sims	bsims		
Jonathon Paul	jbpaul		

## 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.

### 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 15, CS 18 or CS 19 (or have Pascal's permission).

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 Barus & Holley 168. 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 helpsessions held in the evening throughout the semester. Their exact time and location will be announced in class. Helpsessions will be used to introduce students to the tools used in the course, such as Logisim and Mipscope, and to provide helpful guidance at the beginning of new projects. While these helpsessions are not officially required, they are intended to be useful, so students are encouraged to attend.

There will be also weekly advanced topic sessions given by the HTAs and Pascal, at which topics will be covered in more detail than they are in lecture. Students will be able to ask questions that go beyond the scope of the lecture material. Topics covered will depend partially on the interests of the participants. There will also be weekly recitations, given by the HTAs. These are meant to review and clarify lecture material. Neither the advanced topic sessions nor the recitations are required. Both will be held on Wednesday from 5-7, though they usually won't take the entire time. The advanced sections will be held in the Langage Lab (CIT 217), and the recitations will be held in the Motorola Room (CIT 165).

## Assignments

Assignments for this class consist of 10 weekly homeworks, five projects, and a midterm exam.

Homework is (in most cases) assigned each Monday after lecture and is due the following Monday at 1:55 pm, using the CS31 handin script specified at the top of each assignment!

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 Logisim, a circuit simulation program. The remainder of the projects require you to code: the second and third in MIPS assembly, and the fourth and fifth in Java.

The first and second projects will have an interactively-graded design checkpoint worth 20% of your grade for the project. The third and fifth projects will have checkpoints worth a large percentage of your project grade, and the fourth project will have a written design checkpoint.

Project	Out	Design Check	Due
RISC	September 28	Yes	October 11
Life	October 17	Yes	October 28
Maze	October 28	No	November 11
VM	November 11	Yes	November 23
Compiler	November 23	No	December 16

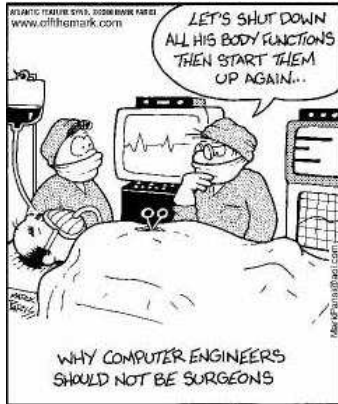
The midterm on October 17 will be closed-book, covering material through Moon-2. You will be permitted one (1) page of notes. You will have three hours for this exam.

## Debugging Policy

If one of your assignments fails a test case, you may receive partial credit for the test case depending on how large the bug is. However, your TAs may not always be able to find the bug. In that case, all is not lost. Your grader will alert you that there is a bug that they cannot locate. You will then have a week to locate the bug and e-mail it to the TA, who will then assign the same partial credit they would have given had they found the bug themselves. You need not fix the bug, but you must explain it well enough so that the TA understands the problem. If you have any questions about this policy, please **mail the Head TAs** at [cs031headtas](mailto:cs031headtas).

## Grading

Your grade for the semester will be determined as follows:



Homeworks (10)	15
RISC	15
Life	15
Maze	15
Compiler	15
VM	15
Midterm	10

Final grade cutoffs will be as follows:

90+	A
80-89	B
70-79	C
Below 70	NC

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.

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 **except Compiler**, 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. Similarly, you must attempt and hand in the midterm exam 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 for the first four projects (i.e., all except Compiler) is December 1. The Compiler project **must** be turned in on the due date unless you have been granted an extension by Professor Van Hentenryck.

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

Absolutely no collaboration is permitted on the exam. 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.

## Course Materials

PDFs of the lecture slides will be made available on the course website for you to print out and take to class if you wish. 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 Logisim (`/course/cs031/bin/logisim`) for circuit design and simulation, and MipScope (`/contrib/bin/mipscope`) for simulation of a MIPS assembly machine. Both programs are freely available on the Internet, so you can download them to your personal computer if you wish.

The TAs will hold helpsessions on both Logisim 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](mailto: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](mailto: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](mailto: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`).