

C S 330 Fall 2012 - CommentsResponses (All Sections): 19 Total

002 Jay's grading system is severely flawed and needs some significant revision. His peer evaluation pits the students against each other and is a joke because it becomes a game of guess-what-jay-thinks. All projects are not graded at all based on whether you completed the assignment, but whether you have test cases for your project (which can be manipulated). Many other flaws, but these are some basics.

002 This class was insane. Jay is a really good teacher, and is very very helpful when asked, but the grading/workload is absolutely too much. Peers in the class have also voiced their concern about their other classes because of the demand of this one. Also, waaaay too much of the grade was determined in the last two weeks of class. As of last week, 65% of the grading was still unaccounted for. My suggestion would be to balance out the grade distribution throughout the semester by making the earlier assignments heavier, and the later assignments weaker.

002 The course work load was excessive and the grading was too strict for not having a clear grading scale or clear expectations. The strict deadlines ensured that you get far away from your computer or this course regardless of what life throws at you, even preventing assignments from being completed early. I found Jay to be extremely smart and very quick to respond to any questions you may have but the responses often felt condescending and not as helpful as they could be.

002 Honestly, I found it to be the worst class of my college experience. I have absolutely no desire to continue learning anything in regards to this branch of computer science. Constantly putting the student's nose to the grinder does not help them learn. Honestly, I highly doubt that the other students in the class have an opinion differing all that much from mine. I mean, honestly, if the class starts out with 30 to 40 people, and by the end of the semester it's down to 7 or 8 people, then there's obviously something wrong in the class structure. Especially when it's a required class and that happens.

As far as what I think needs to be different, I really do think that the entire grading system needs to be thrown out and reworked from scratch. Students are already putting in plenty of hours coding up the projects, and even more writing the test cases. To throw self evaluations and peer evaluations on top of that for each and every assignment is just insane. And it tends to only get the students angry or frustrated, and people are less effective at learning when angry or frustrated, and considering that the point of the class is to teach the students, this is just taking a step backwards.

I'll also have to say that I think that Racket needs to be downplayed a bit. I realize that it is a good language for the subject, but it is also a language that none of the students will ever really use again. Even if they do use the principles from the class, it will be in one of the more common languages. As such, I think that perhaps part of the class could still be taught in Racket, and part of it could actually be taught in a language that the students might actually use in the field. Of course, this being said, I know that Racket is your baby, so I'm pretty sure that things won't change, no matter what people say. But at least I've stated what I think could help.

Honestly, I really think that the main drive in this course really should be to try and pique the interest of the students so that they will choose to continue learning about the subject matter on their own, and even possibly come to benefit the field at some point in the future. This class does not do this. Not by a long shot. Seriously, try and get the students interested in pursuing the field. If the class fails at that then it doesn't amount to anything.

I really do think that some of the subject matter covered in this class could be interesting if it were presented right. I do hope that it can improve.

002 This class was one of my favorite and least favorite classes of my BYU experience at the same time. Let me start off with what I thought went well. The lectures for the classes were generally quite amazing. Jay really knows the material and makes the lectures interesting, more interesting than any other class I have had to this date. Jay was also incredibly responsive when students asked for help via the Google group, email, chat or face-to-face visits. I don't think I have ever had a professor make themselves so available.

Now for a few thoughts about what I think could improve the class. The most important concern I have for the class is the load. I wrote more pages of writing for essays in this class than I did in my English 316 class which I also took this semester. I spent more time coding for this class than I did for either of my other CS classes I also took this semester (312 and 345). It was by far the busiest class I have taken in college. One could make the argument that the extra credit shouldn't be counted into these calculations, but with the rather strict grading policy, the extra credit doesn't really feel like a bonus but more of a mandatory way to compensate for points missed in other places. Reducing or removing some of the essays and a few of the programming assignments (particularly the web assignment) would help.

The peer grading for the class needs to be removed or changed. Having roughly 10% of your grade in the class depend on how well your grading of a peer matches up with the professors grade is debatable, but more importantly the time spent peer grading wasn't very helpful and was more aggravating than anything else. As I have mentioned before, I don't believe the grading to be consistent enough to have a student's own grades depend on the grading others. Changing the peer grading would, in my opinion, be the single most important step in improving this class.

I have previously voiced my opinion at being very frustrated by the inconsistent grading of the two language papers. Giving almost everyone a zero on the first language assignment (not a single person receiving a passing grade), just to try to encourage them to "learn more" is dishonest. I know teachers don't like it when students try to manipulate teachers for a better grade. Likewise, students also don't like it when teachers try to manipulate them to learn more. It might even be more frustrating for the students because the teacher is in the position of power. The course syllabus should more accurately reflect the teacher's desire of having the first paper being more of a rough draft. I felt like I was lied to.

I think TA help sessions would be a nice addition to the course. It is nice to hear the material from a different source. A session in the beginning of the semester, and before the garbage collection assignment would nice. I really respect Jay and do believe he cares about his students but in the beginning of the semester, Jay gave me the feeling that he really doesn't care about students. I think what gave me the biggest impression of Jay not caring and being kind of a jerk is the email interaction on the Google Group. Jay can be very harsh and it feels like he is discouraging questions. I remember reading some of the emails Jay sent to other students and thinking to myself, "Wow, this teacher really is a jerk". I feel like taking a little more effort to present a more respectful attitude towards students in these emails would do wonders for building better relationships with students.

Let me finish by thanking Jay for all of his hard work. I really did enjoy and class and I appreciate his effort in

continually improve the class and helping us become better computer scientists. I won't forget this class anytime soon.

002 Pros about Jay:

Very smart

Easy to get a hold of for help

Designed his class so that you can't get a good grade unless you actually learn something

Cons about Jay:

Sometimes disrespectful in his responses to students

Cocky

Some grading policies are strange and unhelpful such as evaluating a peer and trying to match Jays evaluation
The tough subjects went right over my head in class but I didn't want to stop him in his lecture because I felt like it wouldn't be helpful because either A) It would just go over my head again the second time. B) I felt like Jay would say something rude about me not understanding.

001 I could put a lot of complaints here but I think it all would amount to frivolous whining, with the exception of 2 things that I can suggest to actually improve the class:

1. I was taken aback on that big announcement on like the second day of class "Only a few of you have actually done the survey on the website like the syllabus told you to, and the rest of you are disqualified from extra credit for the semester! Why didn't you do the survey, it was so easy!". I'm sorry if I only filled out the survey 10 seconds after you made that announcement, but at that point I guess I didn't realize how hardcore you were. I was missing out on all those extra points that really could have helped me later in the semester. Was this really necessary? Was it a test to teach us how strict your grading was? And anyways, how is that fair to, say, a student who registers for the class a week late; is he disqualified too? In my mind this whole thing was completely unnecessary. At LEAST add another week or something to the deadline so more people are aware of it.

2. Clearer assignment descriptions! Yes, I've mentioned this before a few times. "But it's all right there!" you say. Yes, in retrospect it appears clear to us, because at that point we understand the assumptions you make about how the program and its functions are supposed to work, things that we have to infer from sparse Racket documentation and little hidden meanings embedded in the assignments. If we all have to come to your office or email you to ask what you meant by some particular phrase or nuance, then wouldn't it be easier if you just told us what you meant from the very beginning? Because if I'm not sure of something, I just make my best guess (or use my own assumptions) and then if I'm wrong I get slammed in the test cases later on. Case in point: The pltweb assignment. I used continuations for handling control within the program, structuring it more or less like the PHP version from the previous assignment. In the evaluation I got crushed because "I did not use Racket's built-in continuation handling for the entire program state, only control". I thought I was using continuations like you wanted me to, because literally the only thing in the assignment description I had to go on was the phrase "using its automatic continuation tracking". The expectations were not made clear, and I suffered for it. I know you want to help students succeed, even when you set the bar fairly high, but I think that'll be difficult if students aren't even sure where the bar is at.

001 Jay has a learning curve. He comes off as rude, condescending, demeaning, and uncaring, but once you

learn he is just short in his emails and you can't take things so personally, the class gets better. Jay did seem to be less rude toward the later part of the semester. I'm not sure if he puts on the hard beginning to set the tone for the rest of the semester, or if he did change after feedback halfway through class. The class is entertaining but challenging. One of the few CS classes that challenged me and made me put forth personal effort to learn.

001 Jay does a great job. His course is labeled "hard" or "frustrating" but I believe this is because students aren't used to having to put in the work to truly learn the material. Most classes are just "code this" or "solve this problem", where Jay expects you to grasp hard things. It is worth the time for sure

001 The class took WAY too much time. The projects were good and interesting but the interest was drowned out in all of the time it required to do the evaluations and the testing. The papers were also valuable but again took an insane amount of time. It was also frustrating to put in so much time and get such poor grades as a result. Overall it felt like there was too much overhead for what should have been the core of the subject matter.

001 Great class - decent work load - I felt overall I learned a lot, some was more broadly applicable than other portions but all were beneficial. Jay puts a lot of effort into the class and it shows. It's also great to get some exposure to some languages and concepts that are extremely relevant but we otherwise miss in the CS curriculum (a decent portion of the things we're discussing are the exact things the most popular languages are modifying, implementing, revising, etc...).

001 I definitely learned a great deal from this course, and respect Jay highly. I did feel that the course work was somewhat imbalanced, with the majority of the points given for a very few assignments near the end of the semester. I would have liked that to have been spaced out a bit more. There were a few times when a student had become frustrated, and I felt that Jay was impatient with them and a bit rude/abrasive.

001 It seems that whenever you are answering a question on the mailing list, and you could either provide a somewhat 'wordy' response or a brief response, you lean towards the side of brevity. Sometimes this makes it hard to extract meaning from your responses. If this is what you feel helps students learn better, then that's ok; just thought I'd point that out.

001 The grading policies did not accurately determine how much a person learned from the course. A student could turn in a project that was correct, but was required to do a self-evaluation to prove that it was correct. Otherwise, the student would get a 0 on the assignment. The self-evaluation is a good idea for the student to demonstrate that he or she can write good test cases, and demonstrate that it works. However, the student may have written a good program, but forgot to do the self-evaluation resulting in a 0. This therefore does not correctly reflect the student's ability to write a program. A more fair grading system would be for the instructor to grade the project to determine how effective it is, then have a self-evaluation whose score is based upon the student's ability to demonstrate that he or she wrote good test cases and can prove that the code works. The best measure of how well the student did, is based on how he or she did on the project as graded by the instructor.

After the self-evaluation on each assignment, the instructor required a peer-evaluation to be completed, which,

if not completed, would result in the student getting a 0 on the assignment. The peer-evaluation did not contribute to my learning, and really was only another opportunity to get a 0 on an assignment. Again, this does not effectively determine the student's understanding of the work because he or she may have written a correct program, but received a 0 if the peer-evaluation was missed. The peer-evaluation may have been more effective if the student was given another student's code to indicate whether it is correct. However, this would be redundant of the peer-eval in which the student demonstrates his or her ability to evaluate code.

In class, most of the lectures were done in Racket, which was difficult because I felt like I was trying to learn Racket the whole time instead of learning about language concepts. Nonetheless, I value the time I have had to learn Racket. However, it was very difficult to follow the lectures when it was all done in Racket.

001 Jay is a fantastic instructor. He is very animated when he teaches, which helps keep everyone engaged. He is very good explaining complex material in a simple way. He is obviously very brilliant, and the code he writes is quite beautiful. When it comes to grading he can be a little intense. I think the way he grades writing assignments in particular could be better defined. Overall, I really enjoyed the course. I was disappointed to learn that this course will no longer be required for computer science majors. I think that those who do choose to take it in the future will prove to be better engineers than those who do not. Knowing the properties and elements of programming languages has definitely broadened my horizons and has made me a better developer.

001 Jay is brilliant but unfair and has a pernicious manner. His work load is astronomical. His grading is ruthless and self proclaimed pedantic. His grading (5% deduction for any flaw--meaning even typos) dose not reflect the work or craftsmanship of any of the five papers, that are five pages single space and research driven. He thrives on lowering a students grade, its exciting to him. Whenever one goes to get help the only feeling obtained is that he is smart and you are stupid. --i'm not ranting i have several personal examples.

The absolute worst experience i have ever had in academia!
--i can't believe this happens at BYU

001 IfI took the hardest English class and hardest CS class I have ever taken and put them together this class still required more work. I have done twice as much work in this class than any other class I have ever taken.

I enjoyed what was taught in class and Jay is an excellent instructor.

The assignments helped me learn a lot, but it was just too much. By the end it got to the point that I stopped wanting to really learn the concepts and I just worried about completing the next assignment. I started hating topics that I feel under different circumstances I would have enjoyed.

I also feel the grading was a bit subjective.

I would suggest keeping similar assignments but cutting them down. This can be done by giving students part of the code already written or cutting down what each program is required to do. For writing assignments decrease the expected number of lines significantly.

I would also suggest creating automated pass off scripts. This will make grading more objective and should make grading easier on the instructor. Also for some programs it would be great to be able to test our program or at least parts of it to make sure it is correct before turning it in. A lot of times I felt I lost points because I simply misunderstood a few technicalities of what the program was supposed to do.

001 Jay gave us every resource he could to help us including his email, gchat, cell phone number, a good amount of office hours, and TAs which are also often available. Jay has a deep desire for us to learn, which once you get to know him, becomes impossible to ignore. I think he has the right idea of how to make us think before we code. I have really enjoyed working with Jay this semesters and learning from him. I do think this class is too much work, between 12 assignments, which often took me far more time than a (usually) weekly assignment should (in my humble opinion), and five five-page, single-spaced papers and a ten-page single-spaced paper. I would have learned more if I had time to focus on fewer assignments with the writing. I learned the very most while doing those papers though, so I hope that at least the large paper and one or two of the small ones remain. I do think the overall load of the class should be a little lighter. I think I would have learned more that way. Jay, thank you for teaching us, and even for coming to teach at BYU. I really learned a lot from you.

001 The work load is too high for a 3 credit class. Either cut down on the papers, or programming, or make the class worth more credit.

The grading procedures are harsh and destructive to the learning environment.

The 3 day turn in processes per assignment is overkill.

Instructor Name: Jay McCarthy
Course Name: C S 330 Sec-001
Response Rate: 18/20 90%

I am learning a great deal in this course.

Average Rating 6.5/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	7	39%
Strongly Agree:	2	11%
Agree:	4	22%
Somewhat Agree:	4	22%
Somewhat Disagree:	0	0%
Disagree:	1	6%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

Course materials and learning activities are effective in helping me learn.

Average Rating 5.8/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	2	11%
Strongly Agree:	6	33%
Agree:	3	17%
Somewhat Agree:	4	22%
Somewhat Disagree:	1	6%
Disagree:	1	6%
Strongly Disagree:	1	6%
Very Strongly Disagree:	0	0%

This course is helping me develop intellectual skills (such as critical thinking, analytical reasoning, integration of knowledge).

Average Rating 5.9/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	4	22%
Strongly Agree:	4	22%
Agree:	4	22%
Somewhat Agree:	2	11%
Somewhat Disagree:	1	6%

Disagree:	2	11%
Strongly Disagree:	1	6%
Very Strongly Disagree:	0	0%

The instructor shows genuine interest in students and their learning.

Average Rating 5.9/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	5	28%
Strongly Agree:	6	33%
Agree:	2	11%
Somewhat Agree:	1	6%
Somewhat Disagree:	1	6%
Disagree:	0	0%
Strongly Disagree:	1	6%
Very Strongly Disagree:	2	11%

What is going well in class? What contributes most to your learning?

Lectures are very informative. I learn the most from that.

Having paper correlate with class lectures. Its effective to lecture on a subject and then have me write about it. I feel like it really starts to sink in.

Class lecture is good. By the time the paper on a subject comes due I don't have much trouble coming up with five pages of material.

Lectures are very good. I end up understanding pretty much everything needed for the assignments. Site and assignments work really well.

I've never had a class like this before in my life. I can understand why students label you as a terrible professor teaching an impossible class. I think that's partly just intimidation, due to how scarily magical you are in this subject. I don't really consider you a pedantic jerk anymore; on the contrary I respect you and I'm glad you do what you do, and that you encourage students to stand up on their own even after they've been knocked down.

I think my attitude towards this class can be a convoluted analogy to Agent Smith's progression of "I hate the Matrix" -> "I love the Matrix" -> "I am the Matrix".

Week 1: "Argh I hate this class why is Scheme such a dumb poopy language, I've never even heard of functional programming" *kicks chair across the room*

Week 4: "Hey you know this functional programming stuff actually explains a lot of things about recursion and scope and little things I never knew about"

Week 8: "I have ascended to the next level of ultimate knowledge and will soon have the power to disassemble any language with a single glance"

What made the change for me was definitely the writing assignments. Somehow or another, I decided "I'm sick of getting all these points off on my assignments and feeling like a flaming wreck of blackened misery and eternal loathing. Oh, you know what? It's a challenge now. I'm going to PROVE YOU WRONG by getting a good grade on these essays JUST TO SPITE YOU." I'm not sure if this was the right kind of motivation for learning you expect from your students, but whatever, it worked for me. I was soon led on a bizarre odyssey to acquire as much programming knowledge as I possibly could so I could use it in my essays. That kind of open-ended research made me start to like the stuff a little bit, and at this point I'm just kind of coasting along on that momentum. I was just reading about how to use monads in Haskell and how a type system could accommodate them. I was also putting together a huge chart of every significant programming language I could find and how they relate to each other. It's a work in progress.

So that's my experience in a nutshell. Yes, this class is challenging and it addresses a lot of new concepts that I've never even heard of nor considered before, which kind of leaves the mind boggled - but heck, that's learning. So I say just keep on doing what you're doing.

Frantic research online is what contributes most.

I feel like the lectures are very helpful and contribute to my learning. I have a mind-blowing experience just about every lecture. And though I don't love the writing assignments, I definitely learn a lot from them as I research the various topics.

Being able to see my potential grade is the only thing that makes me want continue with this class.

The exercise of researching and writing for the language papers makes me learn more about and better understand the topics. I feel I have learned more from these papers than any other assignments.

The examples you show in class, especially when you explain how they relate to programming languages in general.

Class is going well. The homework is helpful. I question the need to all of the papers we need to write in this course.

Ever since Jay started using the whiteboard it's become much easier to follow. We've been talking more about principles rather than implementation details and it's been much easier to get more out of the lectures. Following Jay's fast-paced coding makes it really hard to understand the principles we need to be learning.

I think the lectures that have been done away from the computer have helped me understand the most lately. (garbage collection lectures)

Thank you for going over the last evaluation. I found it very helpful for you to address some of the concerns students were having. It also helped explain the importance behind the essays, something that was difficult to understand at the beginning of the semester.

The whiteboard is nice

We have started doing more lectures on the whiteboard, moving a little more slowly and thoroughly

through the material. I like this more than the previous portion of the course because I personally learn better this way. Jay makes himself incredibly available for help (office hours, phone, email, chat, etc.) which is helpful as well.

I learned a lot from my research while writing the language analysis paper.

I am learning the most from the writing assignments, especially the language evaluation. It was really cool to get under the hood of a lesser known language to find out what interesting things they are doing.

I also really appreciated that Jay modified the format of his teaching in response to feedback.

What could be improved? How could this course be more effective in helping you learn?

A little more help on the web stuff. I know we were supposed to learn web stuff on our own if we hadn't done it before, but that didn't work very well for me...

Creating a Learning Environment

1) Never pit student against each other

--you encourage me to drop my peers grade so mine can increase, this is wrong!

2) 5% for grammar and language is absurd! You are accomplishing nothing but infuriating students. Because the continual bombardment there is no faith to complete assignments---whats the point i wont get due credit for it. PS you justify this is saying there are only a few struggling, no you have yet to give out a valid 100 on a paper...this is because their peer was polite or you missed something! 5% for any other programming/concept/logic flaw should be heavy.

3) You look for reasons to vindictively dock people.

-- "read my first paper... it got a zero"

-- taking about formatting issues..."this deserves a zero"

-- "and now your at a zero"

Part of learning is the environment...what are you inspiring? It could be so much better.

Not much comes to mind.

Now that we understand how assignments and papers work it flows a lot better and I feel way less stressed, but before we knew it could have helped to have some more direction, maybe examples of good and bad papers.

With regards to the criticisms noted about the class in the previous survey, I'm tending to disagree on a lot of those points (and agreeing more with the defense you made to those criticisms when we discussed them). The only thing I could think of is: do the self-evals and peer-evals need to be done sequentially? I don't see why the "homework turn-in workflow" can't be reduced from 3 days to 2 by allowing all the evaluations to be done concurrently. As long as the turn-in deadline is the same for everyone, that shouldn't be a problem right?

Give examples of what he wants on papers in class. Cover some of those points that could be in the papers.

I feel like it is very effective already

The work load is too high for 3 credit class. One of the 10 page papers is equivalent to all of the writing I did in Eng316 (Technical Writing).

Class procedures destroy a constructive learning environment. For example, 10% of my grade is based on a guessing game. Moreover, in order for my grade to improve my peer's grade must suffer.

I am not learning real critical thinking and analytical reasoning skills. This is because my time is spent trying to guess what curve ball Jay is going to throw next; I'm only developing my "Jay-dar" skills, which I feel is rather useless in real life.

Although I feel I have benefited from the writing assignments, I have often wondered if their length was perhaps a little on the long side (especially when writing one after the other later in the course). Would a student be able to learn just as much and be more direct and to the point in 3 pages instead of 5? I think so.

I have never written so much in one class as I have for this one, and we still have several writing assignments left. To write polished essays takes a large amount of time and I find myself spending 2 to 3 times as much time on assignments for this class than any of my other classes this semester (C S 345, C S 456, and C S 465).

I think the writing assignments are valuable, but I think shorter writing assignments would have the same effect.

I'm not sure as I feel I am learning a lot.

It takes WAY TOO MUCH TIME. The time I'm spending in this class is causing me to do poorly in my other classes and I'm not even doing well in this class. The principles are important but there has to be a way to scale back the amount of time required. For example I've learned a lot from the papers but mostly from the research required, not the actual writing of the paper. Could it be possible to find another way to get us to do the research like a quiz or something?

I think maybe talking more about the concepts separately from the code base helps a lot. It would be nice to even talk concepts at the board for 30 minutes and then show code examples for 20 minutes. That can't necessarily be applied to every lecture, but as a general principal I think it could be helpful

I feel it would be more effective to have a couple little assignments that just help solidify some of the concepts. When I went to do the Racket web programming assignment, I wasn't a hundred percent sure where to begin. It would have been more helpful to have a small assignment to guide me along (maybe something along the lines of what we did in class but more catered to the web assignment). These wouldn't be big assignments--maybe not even graded--but there for students to do and check if how they're doing the big assignment is correct.

Now that the grading on the written assignments is more lenient I am happy

I think if we had fewer coding assignments and more assignments to research topics, like we've been doing with papers, I would find the experience more valuable. I like the papers that Jay has added to the course, but with the programming too the course is a bit heavy.

I'm not good at taking notes, and it would help me a lot if there were any sort of notes or lecture slides that correspond to our lectures we're doing on the white board now. Previously I could look over the code Jay wrote in class to refresh my memory as I did the homework assignments, but it's harder now that that's not an option.

I think that some group assignments could be extremely valuable. Perhaps for a writing assignment, if we were assigned to groups of 3 or so to collaborate.

What course resources (i.e. code notes, TAs, professor office hours, professor phone calls, etc) have you used? What else should I provide?

Code notes, the class website, and links from the class website. I think plenty of resources are provided.

Everything!

I've used the notes from class discussion posted on the class website. It would be nice if we could get a screencast going again, like in previous years.

Visited prof, watched google group. I think everything we need is provided.

I've used the code notes in particular to mimic what you did in class for the applicable homeworks. Most of my learning outside the class has been from wikipedia, linking to primary source articles in the ACM, in blogs, and other CS doctorate pages, and maybe a dissertation or two. I think it's good how it is.

I have used Google group.

I use the code notes and the Internet. Seems to be enough for me.

I use the google group and personal emails.

I have used lecture notes, the referenced readings (occasionally), and the Google group. I can not think of a new/different resource that would be useful.

Usually I use the email list and that is sufficient, but once I went to your hours and that helped a lot. I think what you offer is sufficient.

I mostly use the message board you set up. I haven't had the need to approach the TAs or you.

Professor office hours

For me, I think there are lots of resources to accommodate many learning styles.

Code notes, Google group, professor email

Maybe more code samples

I visit his office, email, chat, and visit his class notes online. When the old lectures were still online, I used those too.

See answer to question 2

prof. office hours, google groups, course notes

Do you have to wait long for help either from me or TAs?

Nope! Very fast responses from professor (I've never contacted the TA).

Never visited with the TAs. You no.

No, the professor has been very responsive.

Nope

I've never tried

no

No. Jay is very prompt at responding to emails and grading assignments

Response time is quick, but the responses are most of the time less than helpful. This is because Jay has the attitude that if he helps us he is depriving us of some sort of learning experience.

No - responses are as prompt or more prompt than any other college course I have taken.

no

Not that I have experienced

no

no

No. I always get a fairly prompt response.

No

The only time I've ever had to wait was when Jay was working with a student already and I wanted to speak to Jay. There are always TAs available. I've never had a class with so much help available at all times.

N/A

not at all.

Instructor Name: Jay McCarthy
Course Name: C S 330 Sec-002
Response Rate: 10/10 100%

I am learning a great deal in this course.

Average Rating 5.8/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	2	20%
Agree:	6	60%
Somewhat Agree:	1	10%
Somewhat Disagree:	0	0%
Disagree:	1	10%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

Course materials and learning activities are effective in helping me learn.

Average Rating 5.1/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	1	10%
Agree:	5	50%
Somewhat Agree:	1	10%
Somewhat Disagree:	0	0%
Disagree:	3	30%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

This course is helping me develop intellectual skills (such as critical thinking, analytical reasoning, integration of knowledge).

Average Rating 5.3/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	0	0%
Agree:	7	70%
Somewhat Agree:	0	0%
Somewhat Disagree:	2	20%

Disagree:	1	10%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

The instructor shows genuine interest in students and their learning.

Average Rating 5.4/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	1	10%
Agree:	4	40%
Somewhat Agree:	3	30%
Somewhat Disagree:	2	20%
Disagree:	0	0%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

What is going well in class? What contributes most to your learning?

Papers force you to learn.

I am learning new programming concepts. However, this class is very time consuming. He should understand that we have other classes and we also have life outside the school. I have been spending about 40 hours a week for this class.

The posted coding examples

Going over code concepts and reasoning about them. I've really liked the recent assignments, because for the first time in the class I feel like I'm learning more about the course material than I am about black-box testing. Please treat the garbage collection programming project this way also.

The papers have actually helped me learn more than anything else.

Jay knows his material well.

Lecture notes, PLAI, HtDP

Honestly, I'm just getting tired of the class. I come and listen to the lectures and learn what I can so that I can get through the class, but any interest that I may have had in the subject has completely died during the course. The information needed for the class is in the lectures, so I guess that counts as something good.

I feel like I am learning more from class during this garbage collection section than I have in other sections. I don't know if that is because you are teaching this section better or more clearly, or if it is just because it is an easier concept to understand. Doing the assignments helps me learn the most.

A lot is going well. I am learning a lot both in class and out of class. The essays really make me understand what I am writing about.

What could be improved? How could this course be more effective in helping you learn?

Your answer fishing could be toned down a little. It's obvious that you have a point to make, but keep expecting the class to give suggestions on what that point is. I'm not opposed to in class participation, but sometimes these fishing expeditions feel like they drag on a bit too long.

Decrease course load. In other semesters, most projects were group projects. This semester all projects are individual which is tough. Furthermore, he has even made projects tougher.

The writing assignments and the amount of test cases needed are ridiculous

The writing assignments could use some pacing. I'm fine with writing 5 page papers, but with the standard that you're holding them to I'd appreciate being given more than a week to research, write, proofread, and revise it. The small amount of time given really cuts down on steps 3 and 4, and if you'd like us to revise our papers more, we simply need more time to work on them (posting comments about the lack of revision on the google groups doesn't help).

I feel like I have learned a lot more since we dropped the computer and picked up the pens. It was often very easy to get lost in the sea of code and then very quickly it was hard to keep track of what was built into racket vs what you coded up yourself.

Also, even with the extra credit, grading is harsh (mostly on programs), and I think that is what gives this class a negative stigma cause other than that, I enjoy to the content and (as much as one can) the assignments.

There are still errors with the self evaluations. If a student complete a requirement, but fails to have evidence, the student has two choices - argue using English, or forfeit the points. There is no way to know if

Emacs allows for fast code production, but during instruction I wonder if I might better follow concepts at a slower pace . . . Just a thought.

This is an introductory course for programming language concepts. As such, I really think that there needs to be some focus on cultivating the student's interest in the subject instead of making sure that the student knows every microscopic detail. Even with cramming in as much as you do into one course, the typical student coming out of the course knows altogether too little to be of benefit to the field, and if the class dropout rate can be used as an indicator, anyone that goes through the course has almost no desire to continue learning about it after the course is over. That would be the class's biggest failure.

Instead, the class needs to help the students gain an interest in the field, so that they will continue to study it even after the course is over. At this point, I find it highly doubtful that any of the students in class will do this, which is disappointing because it is a field that has great potential.

You are very strict on grading and turning things in on time and through your system. This is all fine, most other professors aren't very much more lenient. The hard part about it is your attitude towards the student when the student messes up. What I mean by that is that your responses can seem

degrading to the student. You may not mean for them to be this way but they sometimes come across this way. For me, this makes it harder for me to learn from you because I don't want to learn from a teacher who makes their students feel dumb.

This class takes up so much more time than any other course I have ever taken in college. In particular, the essays take up a lot of time. I know that I am a slower writer so this might be particularly bad for me. From my understanding, the university suggests three hours of homework per week per class credit hour. I, however, am normally spending well over the 9 hours on homework per week for this class, and I have never really been close to this recommendation for any other class.

I didn't find the rawweb or racketweb assignments particularly helpful. I believe the purpose of them was to allow us to see the advantages of continuations. For me, while I saw the connection between the assignments and continuations, it was neither particularly insightful or helpful. I have a fair amount of web experience and these assignments still took me several hours each. Essentially what I am trying to say is that I don't feel the benefits of these assignments are big enough to justify their inclusion into a course which already has an abnormally heavy load. I think an assignment where we use continuations directly (instead of using rackets web constructs that use continuations) would probably have been more helpful.

What course resources (i.e. code notes, TAs, professor office hours, professor phone calls, etc) have you used? What else should I provide?

code notes

He over content super fast. Its very hard to follow. I know he says he can help outside class. However, it is not always possible to take these routes.

Mostly I've just used the code notes and the google group I can't think of anything else that could help

Code notes, google groups, etc. Maybe some graphics would be nice explaining the gc procedures. office hours.

I've visited Jay a few times, and saw the TAs a few times. My biggest resource is probably the code notes.

Notes / Text

I have mainly used the course lectures and code notes.

Code notes, google group

I have used class notes a lot. I can't think of any other resources that would be helpful.

Do you have to wait long for help either from me or TAs?

haven't tried

We have been able to get help.

No

Nope.

No

Nope. That's another strength of this class.

No

No.

no

The response times for this class are fantastic.

Instructor Name: Jay McCarthy
Course Name: C S 330 Sec-001
Response Rate: 25/26 96%

I am learning a great deal in this course.

Average Rating 5.4/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	2	8%
Strongly Agree:	2	8%
Agree:	10	40%
Somewhat Agree:	6	24%
Somewhat Disagree:	2	8%
Disagree:	1	4%
Strongly Disagree:	1	4%
Very Strongly Disagree:	1	4%

Course materials and learning activities are effective in helping me learn.

Average Rating 5.4/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	3	12%
Agree:	12	48%
Somewhat Agree:	6	24%
Somewhat Disagree:	1	4%
Disagree:	3	12%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

This course is helping me develop intellectual skills (such as critical thinking, analytical reasoning, integration of knowledge).

Average Rating 6.0/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	1	4%
Strongly Agree:	7	28%
Agree:	10	40%
Somewhat Agree:	6	24%
Somewhat Disagree:	1	4%

Disagree:	0	0%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

The instructor shows genuine interest in students and their learning.

Average Rating 5.7/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	3	12%
Strongly Agree:	5	20%
Agree:	8	32%
Somewhat Agree:	3	12%
Somewhat Disagree:	4	16%
Disagree:	1	4%
Strongly Disagree:	0	0%
Very Strongly Disagree:	1	4%

What is going well in class? What contributes most to your learning?

Jay going over examples and answering questions is the most useful

Assignments are well paced to help students learn what they are supposed to learn.

The exploratory and participatory learning that goes on in classes and with homework assignments contributes most.

As a student I can tell that Jay really enjoys teaching. He is very dynamic and that helps me to stay focused on the lecture. I think the subject material is very interesting and I like the way that it is presented.

Class is entertaining.

Explaining concepts without code, instead of demonstrating it with code.

The lectures help a lot with doing the assignments, either as providing a starting point or showing us how to do what is wanted in the assignment.

Examples of code typed up in class helps a lot.

The coding in class is very helpful. It is good to see examples of racket code.

I think it is valuable that class time is spent in actual code.

Lectures are interesting, and Jay's enthusiasm for the subject helps.

The most helpful part of this class is the office hours. I get good information out of lecture, but Jay makes himself incredibly available for help and is always helpful when I go talk to him.

Class is great. I most enjoy how willing Jay is to hear comments and suggestions and actually

implements changes or alterations if they are indeed valid and well thought out.

being able to review the code jay writes in class

I like the lectures where we learn about the features of programming languages like laziness, dynamic scope, etc. I think doing examples in class helps quite a bit.

The programming concept is going well. The homework review is the most helpful source for me.

The homework is what helps me learn the most. The lectures feel a bit abstract and what is clear to Jay is not readily apparent to me.

...

I feel like I'm learning a lot in this class. It's helpful that you code and show actual examples in class.

I feel class time is really effective - Jay does a great job of walking the class through a problem, showing us how and why a programming paradigm came about.

homework?

It is well structured and very organized. Feedback is given quickly after assignments and the expectations are clear.

A lot of examples are shown to help us better understand how to do the projects.

I do like when we actually talk about aspects of programming languages themselves like scoping and co-inductive recursion and stuff, that's where I feel like I'm learning the most, rather than just "today we're learning more racket". Granted, all of the racket stuff is just a gateway for us to talk about the deeper language stuff, so I understand its necessity. And as much as I dislike the assignments, I do typically learn a lot from them.

The Homework Contributes most to my learning

What could be improved? How could this course be more effective in helping you learn?

Sometimes clarification on what the purpose of the assignments are for would be nice. At times it can feel like you are simply trying to replicate things you have seen Jay do in class (like for the parser and its additions). Other times, like with laziness, the concept is clear. It would be nice to sometimes have something explained like, "And the reason we are doing this is to show you how _____ concept works"

Sometimes the professor can seem to uphold double standards, saying that he's interested in important parts of the assignment and that he won't sweat the details, but then withholding credit for a question due to an obvious typo.

It would be nice to have more clear learning objectives to each class. I sometimes leave wondering what I was supposed to learn and if I learned it.

I don't feel like there is anything that is keeping me from learning.

More clear directions. Using terms that aren't clearly defined and the wondering why nobody understands what you meant is ridiculous.

Explain a "formula" for writing an interpreter.

Also, we should not have to grade other people. I think it's a waste of time.

The lectures show us how to produce the assignments, but we are graded on test cases. Even if we understand the principles of the assignment and code it correctly we could receive very poor grades if we didn't test very specific things. The wording of questions in the assignment evaluations were confusing, but have become more clear. It was mentioned that lecture in class will be changing, but currently I don't feel like a good connection is being made between Programming Language concepts and our assignments. I like not having readings, but without readings and with lectures just being about code, I don't always understand what concept we are learning, why we are learning it, or what will be important to know for tests or for our futures.

Some topics are confusing and go by too fast. For example, toward the end, interpreters got pretty confusing.

Give detailed feedback on grading if the self and peer grading is different than yours.

Time spent talking about Jay's specific ideas on testing. Assignments feel like a very twisted game of russian roulette. There is nothing more frustrating than to spend time on a project and then to play a vindictive guessing game.

In class: name variables with proper good descriptive names from the very beginning.

I feel like the writing assignments are just busy work. The coding projects are were the learning is. Moreover, there is more writing required in this class then the English classes that I've taken. I really feel like its a waste of my time.

Up until this point in the course I haven't really known what the principles are that I'm supposed to have mastered. I just know what the next assignment is and I do it. I think that's largely because we've been ramping up to being proficient in Racket and now that we're on to topics like lazy languages, I know much more of what we're supposed to be taking away from lecture instead of just more developed racket skills.

At this point in the course I can't really think of anything. I have been very impressed with the organization of the course and the preparedness of the professor.

more directions of what the papers are supposed to be about

I feel that for the first few weeks, the course was mostly a "how to use Racket" course. I think having the TAs have some hours to answer questions and help with projects would be helpful too.

May be another languages other than Racket. Don't get me wrong, Racket is really interesting. However, it has a different approach compares to other languages and that sometimes makes it harder to understand the concept.

I bit more explanation as concepts are expressed. Jay understands a lot anc easily jumps from 'a' to

'c', it would help if he could lead us through 'a' to 'b' to 'c' more.

I really feel like that we aren't learning about attributes of programming languages. So far it only appears that we've only talked about certain aspects of the racket programming language that can be used to make an interpreter.

The class also feels like, "If I wanted to do [this], i would do [this]." I want to understand why we want to add a specific feature rather than just know we want to add it only because we can. With that kind of thinking I don't really understand what is going on in the program code that is created during class.

Also, try and be respectful in some of your responses to other students. As much as I appreciate the outside help through the Google group, some of the comments that I've seen made towards other students seem a little uncalled for if they truly don't understand. On a specific occasion you said something along the lines of "I can't believe you would think that way" and in a recent post you said to a student that if there was something not being perfectly understood that he should man up and just take the 0%. A lot of evaluation questions are confusing, so try to be patient with us.

Sometimes the assignments are a little ambiguous. It would be really helpful if you could give us some examples that would clarify what you're actually looking for.

The evaluation questions for the projects seem like they could use some tuning up. For example if in a project we write 10 functions there will be a number of questions that are the same for each function -- "classes" of questions that could be asked about a function; this is especially true for test cases.

The issue is that if I forgot about a certain class of test case, I get pounded for it 10 times. This isn't to say that I didn't deserve it: I did. But This method of evaluation doesn't contribute to my education as much as may be.

I believe that this problem could be alleviated by a more incremental grading system, or by publishing the evaluation criteria ahead of time.

For example, one could submit the first function to be evaluated, got through the questions, recognize what classes of test cases you forgot, and then fix the rest of your code. I maintain that this would result in more learning.

Also, the plethora of evaluation questions makes it intractable to ask you to write an in-depth reply to each one we got wrong. To some extent, this is unnecessary, but in other cases I would appreciate some guidance along with the "No, you got this wrong".

This could perhaps be solved by having less questions...Or if it were automatically graded (by samuri) that would give you a lot more time to go through after and give advice.

To discuss the last questions [5-7]

I am learning a great deal in this course. Course materials and learning activities are effective in helping me learn. This course is helping me develop intellectual skills (such as critical thinking, analytical reasoning, integration of knowledge).

It seems like the assignments have a lot of busy work involved. Now you might say 'it's designed that way. You have to learn to do the busy work and the multitudes of unit tests' and that's fine, I'm certainly learning that...

But I would appreciate if there was more...I don't know. The demonstrations you do in class are where I think I learn the most *new* stuff, like when you produces the Y combinator, to taught about the theory of laziness.

To be sure, the written assignments are designed to foster that kind of learning, but I think we students would benefit if there were a little more direction. Perhaps if you provided some more reference materials? I don't know.

It could be easier for us to understand what's expected of us

Not allowing late work is extremely annoying. Since each assignment consists of 3 parts and missing either of the first two by mere seconds causes you to get a zero on the assignment, it can be really frustrating when you do a lot of work and don't get credit because you spaced the deadline.

I would like to hear more about actual language features than seeing how to implement them with Racket. Being more kind when questions are asked, even if they have been asked before.

I used to think Jay was a pedantic jerk (and I still kind of think he is, this is not a huge secret) but you know, sometimes we need a good rigorous class that punches us in the face, over and over again, until we slowly build up a resistance to facepunching. This is that kind of class for me, and in that sense it's okay. I do sometimes wonder if everyone else in the class feels kinda this same way, and also I wonder if they're all doing as badly as me at the assignments despite their best efforts.

On some assignments I noticed that I created correct implementations that matched the expected output, yet I didn't implement them using the same theories that you expected us to (for example, my lazy (build-table) function precalculated every cell of the table, rather than leaving it as a set of promises), and I lost some credit for this. It's probably my fault for not clarifying things enough during lecture, and so I end up getting the wrong ideas stuck in my head. Could the homework requirements be clarified to fix this? (I guess the solution I'm asking for is "Can you show us the self-evaluation questions before we submit the homework, so we know what you'll test us on?" but that defeats the purpose. This also just boils down to me having to mark "no" on a bunch of self-evaluation questions for things that I didn't think of in advance, and by the time I'm done I feel like a flaming wreck of blackened misery and eternal loathing)

References on the Homework or assignments for where we can find more information if we aren't understanding what is being asked when there are no relevant notes in webpage

What course resources (i.e. code notes, TAs, professor office hours, professor phone calls, etc) have you used? What else should I provide?

The google group is great, and Jay is very reachable for help

Mailing list. I think what the professor has provided is adequate.

I've used email and professor office hours. I don't know what else would be useful.

I have used code notes and google groups notes. I feel like there are plenty of resources provided.

Code notes and Google Groups.

TAs, code notes

All the online tools are great. Website works well. There was a steep learning curve, and it seems like it was a bit of a work in progress at the beginning, but things are working now. Maybe an example at the beginning of class would have helped so we could see what grading assignments would be like and everything.

I've use code notes a lot and the webpage and a little bit of Racket documentation. I can't think of anything else that should be provided; there are plenty of resources available.

I have used the Google group and that seems to be mostly sufficient.

More examples of racket codes for us to look at.

I've used class notes, the google group, and direct email.

I have used the lecture notes and the google group. For me, these have been sufficient.

Code notes (from lecture), professor office hours, emailing the professor and the google group have all been extremely helpful. I've had more resources and avenues for help with this class than most. It is still a challenging class though and I'm grateful for the help because I have a hard time with the subject.

I have used the grading site, the lecture notes, and the class site. All were very useful (and I appreciate the upgrades to the grading site!), I cannot think of anything else I would have wanted.

code notes and the google group

I have used the google group for help.

Code notes, professors

I have used: code notes, google group

I'm not sure what else you could provide.

I have only used the homework submission website and the google group. I appreciate that there is a google group to get help at any time of day.

Code notes.

I've used the google group, prof. office hours. what are code notes?

I use the code notes a lot. I check the google group often because I don't understand a lot of the questions asked

Professor office hours, code from class. I think that's good for now.

I have used lecture notes, the online textbook, the Google Group, and emails to the professor. These have been enough to understand the material.

I have not really used any of the course resources besides the past lecture notes (and those are actually super-helpful, so way to go)

code Notes and google group

Do you have to wait long for help either from me or TAs?

No, help is pretty instantaneous

No.

no

I have never sought help directly from you or the TAs . . . yet.

Normally, I find someone else with a similar problem and read there responses. So far, I do not think I have had to contact anyone directly.

No

No, emails are very prompt.

I haven't asked for help from you or the TAs yet.

No, the responses have been very quick

nope

No.

I've only had to wait to talk to someone in person once and it was for maybe 2 minutes. I did once send an IM that received no response, but all other attempts to get help have been met with very reasonable response time.

No.

not bad most of the time

You usually respond pretty quickly to the google group questions.

No

Not really

Not applicable

No

No, you've been very responsive. Thanks

no

No.

All help has been fairly prompt.

Never tried

Don't know

Instructor Name: Jay McCarthy
Course Name: C S 330 Sec-002
Response Rate: 12/13 92%

I am learning a great deal in this course.

Average Rating 5.1/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	0	0%
Agree:	9	75%
Somewhat Agree:	0	0%
Somewhat Disagree:	0	0%
Disagree:	1	8%
Strongly Disagree:	2	17%
Very Strongly Disagree:	0	0%

Course materials and learning activities are effective in helping me learn.

Average Rating 5.1/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	0	0%
Agree:	6	50%
Somewhat Agree:	4	33%
Somewhat Disagree:	0	0%
Disagree:	1	8%
Strongly Disagree:	1	8%
Very Strongly Disagree:	0	0%

This course is helping me develop intellectual skills (such as critical thinking, analytical reasoning, integration of knowledge).

Average Rating 5.3/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	2	17%
Agree:	4	33%
Somewhat Agree:	4	33%
Somewhat Disagree:	0	0%

Disagree:	1	8%
Strongly Disagree:	1	8%
Very Strongly Disagree:	0	0%

The instructor shows genuine interest in students and their learning.

Average Rating 4.9/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	0	0%
Strongly Agree:	2	17%
Agree:	2	17%
Somewhat Agree:	3	25%
Somewhat Disagree:	3	25%
Disagree:	2	17%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

What is going well in class? What contributes most to your learning?

Going through problems

nothing. he is very rude. he says he is very helpful but he is not. If you ask him to clarify something, he says there is nothing he can do. its more hostile environment than learning environment.

I feel as though the material in the course goes into depth about the subject matter and we are getting all the knowledge we need about programming languages.

The labs contribute the most to my learning.

Going over concrete examples, keeping things relevant to our assignments

Jay is helpful and concerned about student learning

The assignments, while sometimes feeling long and/or tedious, actually do help me understand the material quite well.

reviewing class notes (in conjunction with racket documentation) is where i feel most learning takes place

I really enjoy the in class lectures. The majority of the class sessions are vastly more interesting than any other classes I have had at BYU.

Talking about general topics. Otherwise, its just another class on a specific programming language. I learn most when we are talking about things like "though racket does it this way, here's how this idea is used more generally",
Its done quite a bit anyway, but that's when I take the most from lectures.

Lectures are nice.

You try to be fun in class, which is good.

What could be improved? How could this course be more effective in helping you learn?

Jay goes really fast and is all over the place in his editor. It is very difficult to keep up with him the whole time. And once you're lost you're lost for good.

there is so much work and project takes forever to finish. grading so harsh that we spend all of our time working on projects.

make grading little soft. focus on theory than just things about racket.

The course is supposed to be about programming languages but too often I feel as though it is a class about testing. While testing is important in all programming I don't think it should be take the focus of this class to the extent that it does.

Help us to know more of what test cases will be asked. It doesn't seem completely fair that if our program does what its supposed to do, we don't get credit if we don't prove it in the way Jay asks.

A grading scale that reflects understanding of the material more than complete test cases. I feel I would learn the material better if I could spend more time investigating proper ways to solve a problem than writing a myriad of test cases. As it stands, the grading system gives no return on the "writing good code" investment or even the "effort" investement.

personally I have had a hard time keeping up with what is going on on the screen (since Jay can type really fast) However, I am starting to get used to this. This was a hindrance mainly when I didn't know Racket

Lectures tend to move a little too fast, I get lost.

i don't know if this is intentional, but you often come across as aggressive (or even insolent) in your email / google group responses. a change of tone could go a long way in promoting both the asking of questions and helpful discussion.

also, given the severe standard by which assignments are graded, i'm sure future students could benefit from a class period, early on, where they are taught how to write your desired test cases.

The assignments and strict grading sometimes make me really frustrated, occasionally to the point of hating the material and class. Every time I fill out the self eval, I feel like I am walking through a minefield trying to avoid the traps. I wonder if there is a way to maintain the academic value but make the work more enjoyable.

Teach the class in a more realistic language

Assignments (and what's required to get credit on them), often feel cryptic.

The grading system, in my opinion, is still a broken mess. But I don't see that changing so I'll just work with what I have to.

What course resources (i.e. code notes, TAs, professor office hours, professor phone calls, etc) have you used? What else should I provide?

code notes, google group

i don't know whether there is any TA for this class.

Mostly just the code notes, I've used them and they are great. Also the google group is good. I can't think of any other resources that would be more useful.

I've used the google group and code notes. I plan to talk to Jay to help with my paper.

Code notes, google groups, past google groups, office hours.

Google group and course website.

None.

I have used the google group and the website as a resource.

code notes, google group

Since such a big part of the lessons is writing Racket code on the projector, I wonder if there would be a way we could see the code on our computer as you are editing it. That way we could scroll up and down and reference things without stopping the entire class.

Chat, google group, online documentation

Code notes.

I typically do just fine on my own. I learn the things in class, and that's enough for what I need.

Do you have to wait long for help either from me or TAs?

No

no but i am afraid to ask his help because he thinks he is super genius and expects everybody to be smart.

In this environment, I would be more than happy if I get anything better than B.

Nope jay responds very quickly

No. When I ask for help on the google groups, Jay usually answers pretty quickly. I don't hear from the TAs much. I went and saw them once, but I don't feel the nature of the class lends well to asking the TAs for help. In other words, what I need help with, they can't tell me.

No, I've never been to either place and had to wait

no

No

fast turnaround

Not at all. The feedback on the Google group is incredibly fast.

No

N/A

I haven't used them.