Full Report

Evaluation for 2020 - Term 1 (Fall)

Course Code	Instructor	Response Rate (Respondants/Enrolled)
COMPSCI 1JC3 (C02)	D'Alves, Curtis	78.46% (51/65)

1. Overall for this course, how would you describe your learning experiance?
(*Scale: 1 Very Poor to 10 Excellent*)

11 Students (21.57%) said: 7
10 Students (19.61%) said: 8
7 Students (13.73%) said: 6
7 Students (13.73%) said: 5
5 Students (9.80%) said: 9
3 Students (5.88%) said: 1
3 Students (5.88%) said: 10
2 Students (3.92%) said: 3
2 Students (3.92%) said: 2
1 Students (1.96%) said: 4

Median: 7.00	Mean: 6.39	StDev: 2.3331	Variance: 5.4431	Not Responded: 0

2. The timing and appropriateness of feedback on your progress:
Receiving assignments back in a reasonable time frame, clear explanation of grade
(*Scale: 1 Very Poor to 5 Excellent*)

21 Students (41.18%) said: 4
12 Students (23.53%) said: 3
9 Students (17.65%) said: 2
5 Students (9.80%) said: 1
4 Students (7.84%) said: 5

Median: 3.00	Mean: 3.20	StDev: 1.1317	Variance: 1.2808	Not Responded: 0

3. Independent critical judgement was encouraged:
(*Scale: 1 Very Poor to 5 Excellent*)

20 Students (39.22%) said: 4
15 Students (29.41%) said: 5
8 Students (15.69%) said: 3
4 Students (7.84%) said: 1
4 Students (7.84%) said: 2

Median: 4.00	Mean: 3.75	StDev: 1.1974	Variance: 1.4337	Not Responded: 0

4. OVERALL, how do you rate the value of this course compared with others you have

taken at McMaster?
(*Scale: 1 Very Poor to 5 Excellent*)

19 Students (37.25%) said: 4
11 Students (21.57%) said: 2
8 Students (15.69%) said: 3
8 Students (15.69%) said: 5
5 Students (9.80%) said: 1

Median: 4.00	Mean: 3.27	StDev: 1.2503	Variance: 1.5631	Not Responded: 0

5. The organization of this course:

Progression of learning material, resource availability, professor was timely and prepared
(*Scale: 1 Very Poor to 5 Excellent*)

16 Students (31.37%) said: 4
15 Students (29.41%) said: 3
12 Students (23.53%) said: 5
5 Students (9.80%) said: 2
3 Students (5.88%) said: 1

Median: 4.00	Mean: 3.57	StDev: 1.1359	Variance: 1.2902	Not Responded: 0

6. The instructor's response to students:

Approachability, attitude, availability, well-explained answers
(*Scale: 1 Very Poor to 5 Excellent*)

21 Students (41.18%) said: 5
13 Students (25.49%) said: 4
9 Students (17.65%) said: 3
5 Students (9.80%) said: 1
3 Students (5.88%) said: 2

Median: 4.00	Mean: 3.82	StDev: 1.3070	Variance: 1.7082	Not Responded: 0

7. The coverage and fairness of tests:

Material coverage, mark distribution, difficulty level
(*Scale: 1 Very Poor to 5 Excellent*)

20 Students (39.22%) said: 4
14 Students (27.45%) said: 3
6 Students (11.76%) said: 1
6 Students (11.76%) said: 5
5 Students (9.80%) said: 2

Median: 4.00	Mean: 3.29	StDev: 1.1712	Variance: 1.3718	Not Responded: 0

8. Please comment on the quality of the TA's in this course:

- The TA was nice and he knew what he was doing, however, I could never understand him because he assumed everyone knew what he was talking about but I had no idea what was

going on and that just made me not go to the tutorials.
- The TA was not very good at teaching, rather just proved that he knew how to code in the language which is not very helpful for students who are just learning how to program and write code.
- TA's knew what they were talking about for the most part but did not seem to have a full understanding of the Haskell language and couldn't answer any sort of specific questions which an internet search would also not be able to answer. Fair marking though.
- The Teaching Assistance for this course, personally, was really good. Some of his explanations for the exercises was really good and he made understanding much more simpler, assuming you do watch both the Tutorial or Lecture Content. But sometimes I also felt the one-to-one interaction could have been much more better. An important point also to mention was that he was quick to respond to any queries we had, which included question related to the course, exercises, midterms and assignments.
- The tut TA seemed to know what the content is but lacked engagement and ability to explain his knowledge. The marking TA, on the other hand, has yet to mark stuff from week 1.
- I didn't enjoy the tutorials with the TA as much. I found them to be hard and dry. Not much to understand unfortunately.
- TA is a very good teacher, responds very quickly when only, not only answer the question but also extend the question
- I like this TA, he gave me lots of useful tools for Haskell.
- Worked well, but needed more to complete marking of assignments efficiently.
- TA was very open and informative
- Knowledgable.
- TA's weren't bad however they could have explained things better in tutorials. I felt I learned nothing in tutorials.
- TA was by far the most useless aspect of the entire course
- Could have had more TA's grading was slow.
- The TA was helpful and did their best to help with questions that I had. Moved a little fast though that's probably due to the time constraints.
- The tutorials were ok and got the job done
- Steven was great I think. He emailed back in time. Probably the best part of this course
- No :)
- TA's were helpful when running tutorials and trying to explain ever answer that went though
- TA was ok. Tutorials had a weird structure of just doing exercises and less questions about actual content. Outside of tutorials TAs weren't much help.
- The TA for this course did not really explain answers in depthly, he would mainly just go through the solution without much clarification on why he did things or his thought process.
- The TA was very helpful during tutorials, as well as answering individual question during it.
- I personally found the TA to be very useful as I found his tutorials very informative.
- I was only in touch with the TA that led the tutorials and I have to say he was excellent. He taught me a lot of the practical content that I probably wouldn't have been able to learn on my own.
- I didn't like my TA, purely due to the fact that whenever someone in the tutorials would ask a question, he would answer with 'basically'. We don't want to know if it's basically right, we want to know if it's right or wrong, and why it is wrong if so.
- TA are some pretty smart people. The have good understanding of the content and explained it really well.
- Slow marking
- it fine
- Quite effective and nice guy, helps us a lot in the tutorial, bunch of excellent codes. 10 out of 10.
- Very great. He is so kind to us

- Not very good.
- The tutorials did not feel like live classes at all and felt like a recorded video. It was not engaging. Only solutions to the questions were talked about no examples were given when teaching. The TA did not seem prepared for the classes.
- I only knew one TA and he was ok. I found that he went through the exercises but didn't do much beyond that. He seemed to be better for people who already had a good background understanding of the content.
- The marking on assignments was fair and concise.
- The TA was very helpful and was able to answer all questions I had very quickly.
- TA prepared a lot for each tutorial. The explanation for each exercise questions are clear.
- The TA was OK nothing special just told us how to do it didn't really explain why
- The TA's in this course were very good as all tutorials were very informative and well put together and all questions that we had were answered to the best of his ability.
- I am satisfied with the quality of the TA.
- Very approachable, helpful, and eager to answer questions.
- We never really understood how the TA was grading our assignments in terms of the subjective portion and I don't believe we were allowed to ask the TA assignment questions so we were really in the dark. In terms of the actual TA, he is a very intelligent man but not good at teaching the concepts. Meaning, he undoubtedly knows the material but he lacks the teaching skills to make the content understandable to his students; some of us stopped attending tutorials. I think it would be a lot better if he had a specific lesson plan in mind to explain every step and understand he's teaching a lot of newbies rather than just answering the homework questions.
- It was okay.
- good
- Tutorial TA and marking TA did really good jobs! However, having only one marking TA increases the waiting time for the assignment marks.

9. Please list aspects of this course that you found valuable and should be continued:
- Microsoft teams use is good -discussions in lectures
- Discussion sessions are valuable and are mark boosters.
- The course content that can be applied to all of computer science and not just to Haskell were the most valuable. A lot of time is spent on stuff that i am convinced I will never have to practically use ever again. Also the format of having the lecture videos about the general theory and idea's where the tutorial videos showed the specific applications and uses felt very organized and made it easier to follow along and plan my time accordingly.
- The Discussion sessions probably were the one thing that has made us to watch the lecture and tutorial content on time. The tutorial videos are absolutely A class, and I wouldn’t change one bit about them. Assignment were really fun and challenging, and was, hands down, the best part about the course. Bonus Marks is another aspect I really wouldn’t want to change, since it pushes the student over his/her boundaries. Me personally doing the extra credits for almost all assignments gave me an advantage in understanding more complex chapter that followed on in the later weeks.
- at the time this is being written, not a single thing.
- 1) The YouTube videos. Both for tutorials and Lectures. These are very useful 2) The discussion session is by far the best thing in the course. It expands really well on the topics covered in the tutorial videos and lectures. Prof Curtis really does a great job in making these discussion sessions fun and interesting and easy to understand. 3) Multiple Choice tests. I prefer, and especially in computer science courses to have multiple choice tests instead of having to write code on paper. 4) This is probably the best thing in the course and should be very much continued and that's the amount of extra marks you can get. There is really a very good chance

of increasing marks in this course. For example, the M&Ms are an easy 5 %, the discussion session, all you have to do is not say 'I don't know' which makes that an easy 10%, the extra credit assignments are also nice.
- Giving out the program for evaluation, and solution for assignments.
- Although some guys said Haskell is their nightmare, I really recommend it to beginners, since this language really tell us how to think like a professional programmer.
- Bonus marks
- The clear explanations and youtube videos/lectures were really helpful! they were always available online and i never had trouble finding them (even just searching for them on google)
- Assignments topics were fun and rewarding
- All the foundations of CS that we learn in this course that will stay with us throughout our career.
- The computational theory part of this course was useful and good information to have for basic computing knowledge.
- Haskell was very uninformative and hard to follow
- Discussion Lecture sections, Participation marks
- The discussion sessions engage the class, though Im not sure if they should be marked. Especially online when students are from different countries and they shouldn't be expected to wake up at 4am to go to class.
- The in class discussions
- The online resources, assignments and tutorials
- None of them.
- I feel like the core concepts of computer science and programming, in general, are very valuable to beginners who have just started to program.
- Approachability of professor was awesome. Assignment and midterm formats were great.
- I think the practice problems were very encouraging and helpful in solidifying out understanding they should definitely cater it more towards the assignments and the midterms.
- How the lectures are presented. Instead of a lecture I need to attend, all the lectures were pre-recorded and available to student to watch according to the which week it is in the course.
- multiple choice midterms -Discussion sessions during lectures -M&MS
- The discussion sessions were very well-organized, entertaining, and most importantly I learned a lot of the content from these discussions. I believe this was probably the most valuable and the best aspect of this course.
- I appreciated how the instructor was always available to message on teams, and I liked how the discussion questions worked in live lecture.
- I can only comment on Professor D'Alves. His lectures are very helpful. The format and the style they're given in are very good and the professor is very good at explaining.
- Assignment templates
- nothing
- The tutorial powerpoint we have are just great, show the codes we need in a clear way. Respect and Appreciate to professor, still giving answers to my email at midnight. Discussion setions are helpful and good way to earn marks.
- It was too hard for me who haven't learned any kinds of programming, but the contents were great.
- Pre-recorded lectures
- The pre-lecture videos were very informative.
- Discussion sessions made the lectures more interesting and engaging. I liked how there was lots of opportunity to make up bonus marks.
- The marking scheme and breakdown. The values of assignments, tests, participation are really good
- I think the programming assignments are very useful to understand the coding language further.
- It teaches Haskell which is a difficult language. Also, as I am not a students of the Department

of Computer Science, this course helps me to know more about computer science.
- the tutorial
- I found the live tutorials were very valuable as they were the only time that we had interactive and instructional help with programming in Haskell and the live lectures were also very valuable as they greatly helped to solidify the knowledge that we received from the lecture videos and were also great study notes to help with prepping for exams.
- Instructor Curtis did a very good job in holding the discussion sessions. He explains things in detail and make them easy to understand.
- Tutorial sessions gave us a good understanding of Haskell and allowing us to work on the assignments alone before going to the tutorial kept us engaged and gave me a better understanding of the Haskell language.
- The Professor should continue his system of randomly calling on people for bonus points. That definitely forced me to be constantly updated on course material.
- The class discussions were quite inciteful. The type of questions the Instructor asked helped me understand the materials better and I felt encouraged to learn.
- participation part M & M
- I feel that the contents/concepts taught in this course are very valuable.
- The instructor responded very quickly and nicely.

10. Please list aspects of this course that might be improved:
- give examples before giving an assignment -teach in the lecture? -felt like I took notes for tutorials, lectures and live lectures it was very messy and I did not know what was to be evaluated.
- There are way too many lecture videos to watch. This course is way too time-consuming, as there are lecture videos (4 per week) and tutorial videos to watch before attending tutorials (30-50 minutes long). For someone who is taking this course as an elective, this course has too much work for the mark that you get and for the assignments that you get.
- I think when it came to the lack of Haskell uses until the assignments became a bit jarring and felt unfair. After going 6 weeks without writing any sort of program and then all of a sudden having assignments due every week that had to do with writing programs made it very tricky to gauge how well I was doing and understanding the material until that point. There should be assignments earlier on. Also there should be more focus on the general idea of computational thinking rather than actually working in Haskell, because from what I gather it's not a common language. Also, whenever I looked up supplementary materials online for Haskell to help my understanding, a good amount of them started off with 'Haskell is not a language for beginners...' why have a beginner comp sci course in Haskell then?
- There are a lot of aspect I find that can significantly improve this course. This first is the length of the lecture content. Taking several courses at a time, I felt like the length of the lecture content could be significantly shortened as watching one single chapter for more than an hour would first not really help the student grasp the content properly, second would make it really exhausting, and thirdly to review them just before an exam or a midterm would be very challenging. Midterm exams is another aspect I think that could be significantly changed. Having Haskell question where you have to first hand code it would be very challenging to do it in less than 2-3 mins, since we are actually in an exam situation. I feel like using logical Haskell questions like express termination questions or thinking about how a function terminates is much more viable. Another aspect I would like to point out is that we received one of the hardest assignments just before a midterm, which really made it impossible to do both well.
- Using a better programming language, I get the reason but still, Haskell was equivalent to hell. I could go on a tangent since there is so much but it would probably go to waste so I will save my time and not,

- 1) The time frame between each assignment. We are given 2 weeks between each assignment which at first sight looks good but I found my self to be stuck and doing the assignment on the very last days which could be my mistake but I also have to work on other classes and assignments and with the level of difficulty of the assignments, its really hard to complete them in 2 weeks. 2) The level of difficulty of the tests and assignment. The first test was easy but the second was super hard in an unexpected way. I ended up getting a really low mark which I still don't understand how because I thought I was very prepared for the test.
- Haskell is a pretty hard language, please reduce the difficulty of the assignment or forces more on LEC and TUT
- Dr. Farmer's lecture videos are too theoretical, he could give us more examples. Like how RSA Scheme works and so on.
- Tools to learn haskell, not much direction given in this respect.
- I didn't find the tests to be that nice... then again, i can understand that i would need to study more to get better grades. but they were a tad bit difficult
- Not enough application opportunities for lecture content Need solutions for assignment's extra credit portions
- The teaching of Haskell was not easy, pre recorded lectures were long, pre recorded tutorials were long, basically had no time to watch them most of the time. We kinda just jumped into haskell, did not do any basic learning like I had done previously in another language like python.
- Easier way to learn the course
- It is really hard taking this course online. Having open office hour video calls to go over Assignments would have been useful
- Little more guidance on assignments, maybe even one sample input/output for each function would be great. Right now its just all up to you to get right. Yes, one can presume what the output should be based on the function declaration but it would greatly if we were given even 1 sample input/output.
- The lectures were very boring to watch and uningaging
- It would be cool if the suggested problems each week were in a jupyterhub sort of system a kin to 1MD3 with feedback on test cases and such. This course at time also really feels like an IT course or a history of computer science course, I think it might be interesting to focus more on haskell on functional programming.
- Expectation level. This is a beginner course, so either stick to only theory or programming. Doing both at the same time is difficult and it fells like a lot of content
- All of them.
- I am not quite sure how to answer this question. It's hard to improve a course about the fundamentals as they are generally straightforward
- Learning aspect can be improved greatly. Lecture and Tutorial videos were not enticing and too long. Course content is a bit too challenging for the way it is taught
- I think that there were way too many videos to watch. Each week we had to watch two sets of lecture/ tutorials as well as attending two live tutorial and lecture sessions. I think that it should be cut down to either shorter videos or no live lectures/tutorials. I also think there wasn't a lot of help given for the assignments, they were quite difficult and I found that there was not much resources or people I could go to , to ask for help.
- There must be more model answers given so that the student is able to understand what is the nature of what has been asked by a given question.
- help Provided to students on assignments -the Providing of past midterms to study from - Leniency to students regarding deadlines on assignments (as we are learning in an online environment for the first time)
- I believe there were only two TA's available for this course, so we were not able to receive most of out 'subjective' assignment grades until two or three weeks after we've submitted them. One TA would grade the assignments and one would teach the tutorials. In my opinion, one TA

grading about 90 assignments in a required short period of time is extremely hard.
- There is a lot of content in this class, which is why there are youtube tutorials and lectures on top of live tutorials and lectures, so I think maybe relooking at the syllabus and lightening the load would be a good idea.
- Assignments could be more frequent but easier.
- marking and feedback
- nothing
- Maybe post lecture video earlier, not post it week by week? I think weekly M&Ms should be private, only viewable between students and teachers. For assignment, I think there should be a correct output provided before we hand in the code. We can check if our output satisfies the given output, if everything is correct, (for example, we can have 30 different cases for an assignment, and if our codes satisfy the 30 cases, we can guarantee we get 30/30) we can make sure we get 100%. I mean sometimes we write a function that we thinks works well but still lose mark for detail parts, if we can check our output with the correct output, we can guarantee our mark.
- As I mentioned in question 9, I'm having a hard time on understanding the coding because I don't have such experiences. Somehow it might need more explanation on the learning contents.
- TA
- the assignments are not for first time coders, they could be made easier. -no one except some students ask questions on the teams channel and the answers are not helpful at all. -the assignment grades are not given if not followed exactly by the way the solutions are even if the code runs. -the answers to practice midterms were not Provided hence I could never be sure of my solutions before a midterm.
- There was a lot of youtube videos to watch and some were very long, it would take multiple hours just to watch the videos. I think there should be office hours or tutorials where you can go and ask for help on the assignments instead of spending the whole tutorial on the exercises. For the first assignments it was not very clear what the marking scheme was, and was still not really laid out for the rest of the assignments. I feel like there generally could have been more support provided on the assignments. Very vague explanations about what was expected on the midterms and exams. I did to like how you were unable to go back on questions during the midterm because then if you remembered something later on then you could not edit your answer. I think this caused me to make more errors since I was also under time pressure so I wouldn't want to spend a lot of time thinking about one question, forcing me to only think about it quickly before moving on. I also would have preferred feedback on the midterm or have it shown what was marked incorrectly on the quiz section on avenue so you could see what areas you needed to focus on, instead of just having a mark thrown at you and not knowing what you did wrong. I found the prof was a bit hard to approach and seemed condescending if you did not already know how to do some things, which was a bit upsetting since this was supposed to be an intro course.
- The assignments were extremely hard marked objectively and the information is new and challenging. Especially for non-CS students, it's much too hard if this class was a stand-alone. However, we take 4 other classes which make getting a good mark in my eyes really hard
- I think the tests were a bit difficult.
- I hope there are more recorded videos for both lecture and tutorial. As I have a twelve-hour-difference, it is difficult for me to attend live courses.
- the rate at which this course is done
- Some aspects of the course that could be improved would be the required attendance of the live sessions as although they are great help, having to watch 3-4 lecture videos and then 1 or 2 tutorial videos and then having to also attend 2 lectures and 2 tutorials all in the same week was quite demanding especially when other courses are demanding a lot of attention to be put into them, I felt like I had to sacrifice watching lecture videos one week just to do all my necessary

work in other classes and then I would be behind come the live lectures so I was slightly confused throughout them if I wasn't constantly up to date on all the videos. Other classes either had live lectures or lecture videos, but not both, so the course load in terms of lectures and tutorials was quite demanding.
- The lecture videos were boring and difficult to learn and remember. Dr. Farmer is very rigid when introducing these materials. Many parts of the lectures are almost just reading the PPTs and I think he could do better by giving more detailed explanations and examples like Curtis did in the discussion sessions and the tutorial videos.
- Lots of what is covered in class are not on the lecture slides and there are not very many places to get this information and it always shows up on the midterms. We need feedback on marking because I have not been satisfied with losing marks without an explanation of where. It prevents me from being able to improve and makes me feel like I am helpless against losing marks again on the next assignment.
- There should be a better explanation of the programming language used and more coding examples in lectures to promote better understanding. For the people that have literally never done programming or been exposed to it, it was very difficult to understand and then correctly apply their comprehension to the assignments and exams. This was unfair because the assignments weren't truly reflective of a lot of people's skills and learning; some may have done extremely better if the content was explained better in tutorials rather than just writing solutions live. Maybe additional assignments that have less weighting but probably build beginners' understandings would've been more effective. For instance, prior to any major application based assignment, a better foundation of understanding should be built using minor assignments/tasks that reinforce basic coding formulas/techniques. The foundation of understanding may seem to be already covered by tutorials and their respective homework, however, it's not actually effective for students; especially for beginners online. Subjective marks should also be properly explained, I understand it's 'what makes your code look good' but that's relative and students need more guidance on that. The midterm practice exams also really do need answers even if it's last minute. In reality, they work by forcing students to find answers but that's completely useless if we find or come to the wrong answers. No one can even compare answers for the practice exam because since it's online a lot of us don't know each other and unfortunately never had the opportunity to make friends with classmates and even if some of us that managed to create group chats to compare we have no way of checking if either of our answers are right. Finally, maybe not having any answers for practice questions would have been somewhat more effective if we were able to grasp the content better relating to my initial concern of having a proper foundation of understanding.
- I personally found the assignments to be very challenging. Perhaps the difficulty level could be lowered as this is seen as an introductory course.
- none
- N/A

11. Additional comments:
- I ABSOLUTELY LEARNED NOTHING ABOUT CODING. I'm not screaming just emphasizing. the live lectures were good when we discussed real-life computations but the tutorials were not helpful to do the assignments you assigned.
- This course was very disappointing and made me think if it was worth it or not.
- The Prof was very discouraging. Made it hard to feel motivated after receiving even a mildly poor mark. One of the first things Curtis said in the live lectures was 'if you are not a *computer* person already this course is not for you and you may as well switch out'. Personally, I am not a *computer* person which is exactly why I decided to take beginner computer science classes because it's something I would really like to learn! This mentality and lack of encouragement continued throughout the course, where if you don't have prior knowledge and experience in

computer science it feels like there is a set bias against you! However this did not apply to the assignments and tests and such, just the professors attitude towards students in live lectures. It could be really discouraging. Curtis is wildly condescending and unapproachable. When asking him a simple question and I fumbled slightly he insinuated I was incapable. After messaging him he dismisses your concerns and doesn't answer any questions. After asking about one of my marks as I believe there was an issue with the marking script he told me i 'must've' been running it wrong although i was certain I wasn't and provided no further explanation on how to run it correctly if so. Also, he mentioned he was having a review session and I asked him when it would be on teams because he has yet to post the date anywhere and i like to be prepared and he left me on read without responding. Overall, Curtis has single handedly made me give up on my hopes of minoring in computer science and has deterred me away from ever reaching out to a professor for help. He is not only un-encouraging, unhelpful, condescending, and snarky at times, but also seems to think that only a certain type of student can succeed in his class and he is entirely non-understanding.
- Overall good experience, would recommend it to anybody
- it was an experience and I won't regret taking the course but still it was awful.
- The course is really interesting. You don't only learn about coding in Haskell but you also learn about other things other than coding whihc I really enjoyed considering that coding isn't really my thing.
- Haskell is a pretty hard language!!!!!!!
- overall course was good; small complaints (covered earlier) keep it up :)
- Very enjoyable course overall!
- Great instructor.
- no
- Nope glad its done
- This was my worst course this term not because it was hard, but because there was literally no breathing room. The material was there constantly, and it was very hard to keep up with all of it. It should be a little more condensed. I already know how to program and I thought that this course was challenging. I can imagine that absolute beginners would struggle. Curtis, you should acknowledge that. Also, your teaching methods don't work for everyone and at one point, I thought that your expectations from every student were a little unrealistic. The assignment rubric needs to change - it looks terrible and its hard to understand what you want the program to do. Some questions only had the type signature and nothing else. Also, on Teams, people don't reply to messages - a lot of questions are not answered. It just feels like the course is trying too hard. Again, courses don't have to be stressful in order for us to learn. And this is C02 - not everyone really wants to go into becoming a programmer. I think you were scaring people off into giving this a shot, more than showing how fun and interesting computer science is. Its a first year course. Calm down.
- If I didn't have to take this course, I wouldn't. My favorite part of the day is being asleep, and I wish that I could just be asleep permanently.
- Although I personally did not gain much from this course as I am already familiar with the topics covered, I can understand the importance of learning the fundamentals before more advanced topics
- The test format was a little harsh by not allowing students to go back to questions they answered which makes the student unable to distribute the time effectively between all the questions of the test.
- None
- nothing
- No
- This course does not seem to be suitable for first time coders as it is functional programming and the assignment has upper level mathematics questions and topics. Live lectures are more focused on theory and not the tutorial videos posted, very less focus on haskell during live

lectures.
- Although it is a hard course, I think it is valuable to learn.
- n/a
- This course was kind of disappointing, good exposure, but really disappointing because I went in excited to learn a new skill and I'm sure a lot of other students did as well but it went way worse than expected. I'd say that's an understandable reaction for any student and a difficult course however this is really different. It's different because it feels like it didn't have to be this way or go the way it went. I know it was difficult for the professors and all university staff but I just really hope they recognize a lot of these marks are a again no where near reflective of some of our skills. I know personally for a fact that if this was in person or if the tutorials were delivered a bit better I would've definitely done better. I'm honestly ashamed of how I did in this course because I've never not picked up or understood course content this poorly(in terms of the coding part).
- I love this course!!! but also I have realized that computer science is So HARD!!!
- N/A