STUDENTS' RATINGS/COMMENTS ON MODULE

Faculty:	SCHOOL OF COMPUTING	Academic Year:	2008/2009
Department:	COMPUTER SCIENCE	Semester:	2
Module:	DATA STRUCTURES AND ALGORITHMS - CS1102C		

Qn	Items Evaluated	Module Avg Score	Nos Responded
1	Overall opinion of the module.	3.639	266
2	Grade likely to get for the module.	3.850	260
3	Difficulty level of the module.	4.199	266

QN\SCORE	5	4	3	2	1
Qn 1: Overall opinion of the module.	Excellent	Good	Satisfactory	Unsatisfactory	Poor
Qn 2: Grade likely to get for the module.	A	B	C	D	F
Qn 3: Difficulty level of the module.	Very Difficult	Difficult	Average	Easy	Very Easy

Frequency Distribution (Qn 1: Students' Overall Opinion on the module)

	Nos. of Respondents(% of Respondents)				
ITEM\SCORE	Excellent	Good	Satisfactory	Unsatisfactory	Poor
Module	45 (16.92%)	108 (40.60%)	90 (33.83%)	18 (6.77%)	5 (1.88%)
Module at Same Level (Dept)	164 (12.12%)	615 (45.45%)	459 (33.92%)	86 (6.36%)	29 (2.14%)
Module at Same Level (Fac)	204 (10.16%)	894 (44.54%)	746 (37.17%)	127 (6.33%)	36 (1.79%)

Frequency Distribution (Qn 2: Grades likely to get for the module)

	Nos. of Respondents(% of Respondents)				
ITEM\SCORE	A	B	C	D	F
Module	59 (22.69%)	127 (48.85%)	56 (21.54%)	12 (4.62%)	6 (2.31%)
Module at Same Level (Dept)	394 (29.94%)	591 (44.91%)	253 (19.22%)	62 (4.71%)	16 (1.22%)
Module at Same Level (Fac)	529 (26.93%)	1004 (51.12%)	351 (17.87%)	63 (3.21%)	17 (.87%)

Frequency Distribution (Qn 3: Difficulty level of the module)

Nos. of Respondents(% of Respondents)

ITEM\SCORE	Very Difficult	Difficult	Average	Easy	Very Easy
Module	90 (33.83%)	141 (53.01%)	33 (12.41%)	2 (.75%)	0 (.00%)
Module at Same Level (Dept)	298 (22.04%)	603 (44.60%)	391 (28.92%)	52 (3.85%)	8 (.59%)
Module at Same Level (Fac)	330 (16.45%)	767 (38.24%)	815 (40.63%)	82 (4.09%)	12 (.60%)

Q1. Please comment on the strengths and weaknesses of the module, and suggest possible improvements.

1. Maybe due to complain of the seniors, the module have removed quite a few chapters from the syllabus compare to last time. However, i think the current syllabus has removed too much thing. For example, i think the AVL is n important method for the binary search tree to function effectively, yet it was removed from the syllabus
2. very creative
3. It is properly paced such that i never feel overwhelmed
4. Students who are not familiar with computing are at a disadvantage. Very little emphasis on coding/ programming language. What are students who are unfamiliar with C++ language to do?
5. ample help and guidance is given. the module has too much content that the lecturers have to rush. increase the number of lecture hours
6. Both sit-in labs and take home labs should be graded. This is to allow people who has put in much effort into the take-home labs to at least get some grades for their hard work. This also allows the students not to be too stress up when they go for their sit-in labs. Example some students who could not code the programs during the sit-in labs would be demoralised and thus might lose interest in this module. I also know of some of ECE students giving up this course to go for business courses instead, due to this module.
7. More relevant than cs1101c. An interesting module. But three months is really an inadequate amount of time to master the module, so the lecturers should not have such a high expectation of us in doing well as we are not only taking 1 module, but 5 MODULES.
8. N.A
9. The module is very packed throughout the semester and there are usually many new topics to learn before the old ones can be comprehended. I feel that the labs have improved since the last time as it limits the graded ones to just every other week. Perhaps there could be an extra lab session for the last few topics as currently there isn't to the best of my knowledge.
10. This is a good module to study as it trains one's thinking. However, it is very time consuming and requires alot of practice on behalf of the student. People without prior background will find it very hard to catch up. People who are naturally very good in computing will do very well, while the average student will struggle to survive.
11. This module provides sufficient training in programming skills as well as theoretical knowledge. However, some of my peers have done some research on the length of a standard C++ programming course and it seems that it takes around 2 or more years to train a certified C++ programmer. It

seems to me that even though the course does seem difficult, it may not train us sufficiently to code well enough in a working environment.

12. The labs are a waste of time as it carries too little marks and there are too many free marks that had certainly caused people to spend less time on the labs and instead on other modules that are much more worthwhile. Free marks form too huge a portion eg.marks given for comments, formatting etc. thus even if a student doesn't really study for the lab and with the lab score being best 3 out of 4, he would only be a few percentage points behind someone who had put in 8 hours to do the take home labs. This is certainly one of the main reasons people don't do their labs and I think the professors had overlooked this. Improvements could include 0% for non-compilable programs, no more best 3 out of 4, and at most 10% free marks. In addition, the free marks also extends to the mid term test where weaker students get marks they don't earn and the extraordinary easy PE compared to past years and even our own take home and sit in labs. In conclusion, if students are consistently fed with simple questions and free marks, this will dampen all the student's interest in this module. Lab scores take too long to be released.
13. Its alot of stuff even though it is a 5 mc module. It could be better if some stuff could be taught in 1101 itself.
14. The module was taught very clearly and the continual assessment throughout the entire semester helped me alot in getting familiar with the key concepts. However, I had some difficulty with the labs at the beginning as I was not very comfortable yet with the new concepts in CS1102C. Perhaps for the first lab practice, more than one question can be released so that students can become familiar with the object-oriented programming style.
15. This module allows us to think critically so as to produce efficient and effective programming methodolgy. Because of the numerous topics required to be cover in this module, perhaps some topics could be shifted forward to cs1101c.
16. It was satisfactory for me.
17. Coding in C++ is slightly simpler compared to that in C, but the change of mindset from procedural to object oriented is quite difficult. Luckily the lecturers and tutors are experienced enough to facilitate ease of the transition, and brain teasers are given during breaks to ease the tension.
18. -
19. ->this module has provided an excellent foundation in c++ programming ->insufficient lab questions for practice ->my reference to library material(initially) MAY imply that the topic on classes was not well-taught
20. reduce the weightage of this module. it will be gd if dr soo can take over from dr tan.
21. I find that the lab questions are not mapped directly to the lecture. This varies with the CS1101C which i did last semesester in which i found the weekly labs very comprehensive. It allowed for me to be more confident with coding on the topics
22. too much to study.
23. For additional exposure, lab questions on Graphs and non-examinable topics can be released so that student can get a better understanding of how to implelement the algorithm for those who are interested. Perhaps more lecture/tutorial/seminar sessions should be made available so that one can get a more comprehensive overview of C++. Moreover, after learning so much programming, I still cannot bridge it to the applications I see. The alternative takehome/sit-in lab is a good implementation as it prepares students for the practical exam very well. However, the frequent sit in labs might disadvantage those who are unable to do programming well.

24. Instrumental in helping me know more about the computing/computer science field. Learning data structures helps me to understand some of the error messages I encounter when I surf the net.
25. It is a useful course however too little time is allocated to study it.
26. It appears to me the instructors are always pressed for time. Consequently, we are always inundated with so much information in one lecture, that it becomes an uphill climb for weaker students like myself to keep up. Short of reducing the syllabus even further, I would recommend spreading the lectures over a longer period of time, up to at least three lectures a week.
27. it is difficult
28. Strengths: Teaches us methods to solve problems
29. A very broad introduction to OOP, data structures. Marks are contributed throughout the semester. However, labs tend to be punishing and hard. System used previously in CS1101C is much better in that aspect, with less worry (ie graded labs spread over 9+ weekly take-home labs), instead of bi-weekly sit-in labs with greatly increased stress levels. In addition, it is time to revise on the position of unix-based text-base development environments. IDE systems and CVS have made developing on unix not such a demanded option as those things simply do not work as well as a GUI-based environment. I can handle VIM quite well, but time efficiently greatly increases while I am using Codeblocks + Winmerge (for diffs as opposed to vimdiff) to handle the take-home labs instead. At the barest, the module should introduce how to handle GUI-based environments and the concept of CVS, and better if labs can be done in a GUI-based environment. There should be provision of commented code (especially at the start of the module) instead of simply pseudo-code and codes embedded in lecture notes). Making people copy stuff will not make people learn, especially when it doesn't work and you do not know what is wrong (especially the class and template issues which are new and without examples). This issue is less significant in the latter parts of the module though. On the topic of complexity analysis, there should be more examples in the lecture notes through guiding students how to handle it, instead of just explaining the general concepts of it. If possible, make the revision / help sessions webcast-able also since timetable issues affect the feasibility of going for these lectures.
30. Quite useful in the real life scene for most programmers and engineers. However, it tries to cover too many topics in too little a time.
31. dr soo and dr tan have made this course a very personal experience. they frequently hold help sessions and allow students to visit them for consultation freely. good work!
32. I thought it was good that although some parts were taken out of syllabus, it was still taught in later lectures. Revision/Consultation lectures helped as well. It would have been better if we were taught recursion in CS1101C so that we wouldn't have to grasp this new concept together with the rest of the C++ syllabus.
33. very hard and need much more effort than any other module.
34. this module is interesting but also very hard.
35. even though lecturers claimed that it is for our benefits for not giving us the complete code, i still think some students might be able to learn better if it is given. DIFFERENT STUDENTS have DIFFERENT ways of learning. DON't BE SO SURE THAT BY GIVING OUT code refrains student from solving other problems other than those questions that are covered.
36. The notes were good and the lecturers were good at explaining and working hard for us. They were very sincere. But it was a very difficult module...and it takes very long to absorb and understand each concept. So having sit in labs right after teaching a new concept was too fast and hard to internalise.

The weekly take home labs were also really difficult. Even the sit-in labs were complicated.

37. teachers readily provide assistance when needed
38. very fast paced. hard to keep up.
39. This module does not seem to cater to those without prior knowledge students with prior knowledge of c++ have advantages every tutorials and labs felt like a huge leap from lectures and previous sessions arrange help session (I personally don't mind night class) for those who need it labs are pretty much left to individuals, and at a level that is so much higher than our background knowledge(which is cs1101c) i suggest we have more labs for more familiarisation,and these labs should be discussed fully. (perhaps make this an optional labs for those who need it. less important concepts could be taken out from explanation in lectures more time and effort should be made to more important concepts say like same core concepts from cs 1101c need not be explained at all, very much self explaining from notes (eg, loops) instead spend more time on new concept lke object, template, stl class and for traversal, I feel that concept / workings are more important than result model answer from mid term and final exams should be made available online some webcasts are not available (later part of lectures) and speed non adjustable PE a bit unfair, a lot of my friends have more time extension than us ,(we only have 5 minutes, where as other whose connection were not disrupted can have upto 15 minutes) weightage should be reflected on the question paper, so we can manage our time better
40. This module is too difficult, I do not think i can pass.
41. This module is fun as we are introduced to different things we haven't learned before.
42. I think everything is quite ok.
43. Thus module is a very challenging one and i suggest not to change it !
44. Teaching should be a bit slower since certain parts are covered extremely fast...
45. The strength of this module is that this gives a sound foundation in C++ programming which is much wanted in the industry today, yet this also is a drawback to this module as it tries to puch into the students a lot of important and complex concept in a very short of time. Which only leaves the students understanding the concepts and forgetting it as there is no sink in time.
46. Basically this module is quite packed. This may cause some students to give up after continuous defeat. Anyway, this can improve students' ability in programming a lot.
47. Instead of having to do long take-home labs every week, the sit-in labs help me in managing my time for all the modules more efficiently. I also didn't get a panic attack during PE as it's my fifth time having to code under pressure. Maybe next time the take-home labs questions can be more challenging and different from sit-in questions so that students can learn more. However, please don't increase the difficulty for sit-in as I think it's just nice. Overall, I'm really satisfied with how the module is being administered. The lecturers are understanding and helpful especially with all the help sessions. The technical problem during PE was well handled and I don't think it was unfair to any sessions. Please continue to webcast all the lectures as it's been a life safer for me this time. Whether we were there physically or spiritually, we will all benefit from ur teachings. Finally, I hope the lecturers, tutors and TAs will keep up with their caring attitude for future semesters so that every student will enjoy learning C++ and algorithm, just like me...
48. One possible improvement that I can make is perhaps regarding the release of lab other tests results. The results release takes a slightly delayed time to enable us to reflect on past mistake. Another suggestion is about the forums. Too many forums on the IVLE makes it quite tedious to access. I would like to suggest that "Sharing", "CM" and "General" be combined into one. Overall, I appreciate

the module's teaching methodology.

49. Possible improvements: Provide more practice questions in the tutorials.
50. This module succeeded in teaching concepts, rather than merely syntax, of programming.
51. probably make the class size for lab sessions smaller so that it doesnt look like a mass lecture sort of thing
52. I think the module topics are very heavy and hard to grasp. Perhaps the lecturers could provide more sample codes on top of the pseudo codes in the lecture notes, so the students can have a better understanding on how the exact algorithm works.
53. More time is required to impart knowledge to students 1 semester for an entire modules is really heavy for students who don't have background knowledge about this mod.
54. Bring back the old method of doing graded take home practicals instead. At least it gives us many more practices instead of just 1 before each sit in.
55. Lecturers provides webcast which is essential as this module is difficult. having webcast will enable students to review key concepts periodically.
56. While setting difficult exam questions may urge the students to push themselves academically, what is the point when we perform atrociously for the subject, yet still manage to pass by moderation? Creme de la creme may be identified in such a way, but would that suffice? Can basic competency be assessed too, to give the students some sense that he/she at least understood the basics? For example, can there be MORE questions per tutorial, where they will ask the most basic questions to the most advanced? It will help build confidence in the subject for the students. The elementary questions need not be gone through in the tutorial. Self-esteem is needed in this subject, and I think adding more of such questions will help. Progressive Training is key.
57. Strengths: This module has very good lecturers, lab assistant and tutors. This module enable a student to brainstorm methods to solve critical problems which enhances our thinking skills. Assistance are readily available. Weaknesses: At the beginning of the semester, it is not easy to understand how to code the syntax in sunfire.
58. Can be quite tough for people with no C++ programming background to start on. Should give students more time to grasp the basics of the topic.
59. very hard T_T i think the lecturer should not tell us students not to worry about the language and focus more on the concept because the language is equally important...
60. lab is hard.simply no time to practice
61. Pace is too fast. Should have split the module into 2, or rather spread it over 2 sems.
62. Interesting, but a bit too hard. Could either try to cut the syllabus or make the exams easier.
63. Lab should have lesser weightage if it is marked by different TAs. 18% is simply too large and I have experienced obvious discrepancies in marks from when compared with friend. Eg. Friend gets higher marks from another TA even though code is inferior.
64. Lab should be take home every week, with a smaller fixed percentage.
65. Weakness- Sit in Lab is too tough. Suggest to be easier as at the end of the course, it tests us on the concept learned and not whether we are skilled in programming. Strength - Lecturers are giving extra consultation lectures which is beneficial to me.
66. NA
67. difficult
68. I think cutting down the syllabus to make the module more manageable is a reasonable thing to do.

just keep in mind not to cut down any further, otherwise it will be insufficient for us to be able to use OOP properly. The lab questions are interesting. exciting questions helped me to enjoy myself even if I am in the middle of debugging. haha. Keep it up. ^^ And though CS1102 is not focused in algorithm, maybe it's good if we touch a bit more on which one is effective programs as opposed to not so effective ones, just as guidelines for us when we're programming ourselves. ^^

69. Strength : algorithm taught can be implemented in all other programming languages Weakness : none that i can point out, due to limited knowledge of programming to begin with
70. The strength of this module is that it teaches using C++ as a platform, how to analyse algorithms to come up with the most effective solution to the problem. This cultivates thinking in students. There is not much weakness in this module. It is just difficult to adapt to C++ quickly after taking C because of the difference in thinking processes
71. weakness is too little practice questions with answers are available for student to do. There are only past year papers WITHOUT formal answers provided for students. I think it is quite pointless to do them because we do not even know whether we get it correct. Those teachers always say "you can approached us if you do not know how to do." but it will waste lots of time meeting up with the tutors. AND its even worst if there are questions that we think we did it correct but in fact we did it wrong! Thus i suggest answers to be provided for past year papers.
72. The module is fun to study and the amount of syllabus is just right. Although the questions can be difficult and the preparation for practical exams is quite taxing.Sometimes it can be a bit too time-consuming to study.
73. It would be good to have open book written exams.
74. No answers provided for labs. No answers provided for past year labs. Hard to find a good sample.
75. THe pace is too fast. Every week new things keep coming up and we are expected to master it in one less than one week.
76. It would be better if solutions for labs/sample codes are provided so that students can have an example of how to go about doing if they are totally clueless about it.
77. overall,the module requires alot of time and effort to be put in by the students to practice programming.however,it assumes that the student have a basic knowledge of programming in the first place as the lectures are quite fast-paced from the second lecturer onwards.
78. CS1102C is good in way that students have to think out of the box and understand what was taught instead of just copying lines of syntax from the textbook. however, for students who are not familiar with programming, it takes up too much of their time just to learn the basics of programming. This module requires students to learn many new concepts and data structures which all take time to practice. hence i hope there will be less new data structures in future.
79. too difficult and takes up a lot of time to understand and apply the concepts.
80. Although i can code simple codes after the course, i still feel C++ quite abstract. Have not really getting the feeling yet.
81. Good
82. Mostly clear notes and explanations during lecture. Most fundamentals covered except for files, which I thought should also be an important aspect of programming.
83. The strength of this module is that it really helped me to understand various data structure. The weaknesses of this module are: 1) There are too much contents in this module. How can we learn all the new C++ in such a short amount of time? The worst part is that we have to then apply all the syntax knowledge like a C++ professional. 2) To learn C++ plus data structure in 36 hours in 12

weeks is not enough at all esp for people who are new to C++ like me! 3) I have no idea how the lab tutors are selected. They cant teach. 4) The lecturers really talk too fast and they treat us all as genius like them. They have too many topics to cover!!!

84. Weakness:Too much workload
85. A very difficult topic as known but the lecturers have made a conscious effort in making help readily available.
86. I think the amount learnt is alright. However, this module is extremely catered towards students who have a strong background in computing especially c++ and java. I believe a bridging module or extra lessons to allow students with no prior experience should be provided so as to allow them to catch up on the syntax.
87. It is almost complete, but the teaching could be done a bit slower.
88. The module is qite difficult especially for people who do not have a computer language background. However the new system of sit-in labs are favourable for students. Also a lot of help and encouragement is given by the lecturers tutors and lab TA. I faced a problem during my practical exam, once the entire file ofone of the questions got encrypted and converted to binary language. The lab TA says there were no back up files and so i had to re-write the code again. I think back up files should be automatically created to avoid such problems.
89. Strenghts: 1. It provides webcast lectures so that students can revise the lesson after each lecture. 2. Information given in the lecture notes are very complete. Weakness: It does not give model answer for the lab. Possible improvements: 1. It should provide model answer for labs. 2. Give more practical exercises for the sit-in labs.
90. perhaps more time can be spent on teaching the basics of c++.
91. This module is too tough to be a level one module. The number of lecture hours and tutorial sessions should be increased.
92. We are exposed to quite an amount of critical thinking which is very important. We come to know many new concepts in this module than in CS1101C. But there are too many concepts to be remembered and the exam is a closed book exam which makes it difficult to remember all the data structures available and they way they work.
93. too much too learn in one semester. Have minor quiz to test on student understanding.
94. Very interesting. Good teaching staffs makes the module enjoyable throughout the semester. Reduced course work is now very managable and it is not the nightmares discribed by seniors.
95. Sit-in labs and practical exam can have more balanced marks allocation - they are simply assessments of similar nature, so I don't think it's a good idea to give 15% for PE and 6% for each sit-in lab, especially when the PE is simpler than some of the sit-in labs. Also, marks allocation in mid-term test may be more balanced - more for harder questions, less for easier ones. I suggest giving partial credit for the (very expensive) MCQs as well, or making them less expensive, otherwise it's not the way for students to show their true abilities.
96. The arrangement of this module is good.
97. more code examples are needed, and lecturer can give more examples and output of the concepts covered in lecture notes no need to just read the lecture notes, and for Dr. Tan Sun Tech do not use the camera webcast
98. Good module, provided a wide range of fundamental algorithms
99. labs,PE,and midterms are weight too much.

100. it seems that there is no more space to improve....unless we do not need to learn it....
101. Too much contents are provided such that no time to digest them. In the later part of this course, I was confused by the pseudo codes, which I cannot manipulate with it on the computer.
102. The labs are difficult and there are not enough exercise for student to practice programming skills
103. The content of this module is relatively difficult, and it required more time to learn than other modules.
104. The design of the module is very systematic and clear. However, we do not have enough practices on the last few topics such as heap, hash and graph.
105. a little difficult
106. I still think this module is hard
107. Sit in labs are very helpful~~~~
108. this is too diffcult, especially sit-in lab and PE. This module is going to kill me
109. every point is really difficult.
110. strength: it makes me yearn for the time when i was learning C programming weakness: no
111. I find that I have a better understanding of the whole thing at the end of the semester.At the first half, it's really hard for me.why doesn't slow a bit at the begining.
112. add more practical code for the latter part, and plz provide possible solutions for the past year paper in case we know the answers we get are correct or not, thus we can actually learn...
113. the time is limited for students to master all the concepts. while the lecturers are kind. Also, the grading system of some parts of the tests, like sit in lab, can be unfair.it is subjectively depends on TAs. also, not providing and answers for final exam and no sample paper for mid term is really bad and helpless.
114. I can really learn quite a lot, but the workload is too heavy for me.
115. Explanations of confusing points are satisfying and details of programming is not provided enough. The content of this module should cut dowm a little bit.
116. This module is really very difficult.
117. too many topics are covered but we don't have enough time to digest
118. this module is not arranged well and I think it should be breaked into two modules: one is about c++ and the other one is about data structure.
119. Very good lecturer and tutor.
120. This module is difficult because it needs us thinking carefully and should do large amount of practice in order to learn well. It takes much time to understand the concepts especially for the first half semester. On the contrast, it is interesting to learn this module because it is helpful for programming and algorithm. One suggestion is that we should learn AVL tree which is not in the current syllabus.
121. the most important point of this module is the application， but we actually do not have much time implement what we have learn to real life. that's a pity!
122. A lot of syllabus reduction which makes us learn less things.
123. Interesting course and not as scary/difficult as "assumed" to be. Maybe because much materials were cut down.
124. The code is not really taught in the module, however, we are requested to do sit-in labs and PE, which we cannot possibly do well without knowing the code, and it would take a lot of time if we figure the code out on our own. That is a heavy workload.
125. the module helps me build a solid foundation in programming algorithm. but the content is quite

compacted, 13 weeks seem a bit too short, so many of us find it difficult.

126. Lab questions are not easy to understand, given the way it is phrased. There are also frequent loopholes in questions which would lead to misconceptions, resulting in unnecessary deduction of marks.
127. Provide sample code to weekly lab if possible, so that students may possibly get some tips or pointers from the suggested solutions, in addition to their own code. (e.g. student's code may not be as efficient/polished as the expected one etc) Lecturers are very helpful as they are willing to conduct extra help sessions outside lesson time to help the students, keep it up! The MCQ weightage for the midterm test is too high! 5 marks for 1 MCQ sounds a little ridiculous. Some may find it easy to do and thus the 5 marks is easy to earn, while some may find the MCQ very easy to get 0 marks, as you either get it right or wrong.
128. Too much content.
129. It teaches the programming concept well and was very relevant
130. NIL
131. This module allows us to learn new data structures and algorithms to solve more challenging computing problems. However the use of newly implemented Coursemarker makes the practical session slightly harder as it reduces flexibility in coding style. Furthermore the weekly sit-in labs also makes it more difficult for us to score for the practical component of the module as sometimes its just too difficult to implement a fully workable code within a time limit, I believe it defeats the purpose as the we were supposed to learn how to apply and implement whart we learnt into real time coding, instead of testing our coding speed. Possible improvements would be removing the weekly sit-in labs, instead have a take home lab.
132. I think the practical and sit-in are good to judge the skills but i don't think that the Lab TA's are good enough to mark our work.
133. Please at least provide sample solutions for both the take-home and sit-in labs.
134. Lectures are very dry. Try to make it more interactive and engaging. Provide solutions for lab excercises as it will aid in the understanding of the language and know what is applicable and what is not. Just checking through coursemarker only check whether the outcome is correct or not. It might be better to provide an answer after the lab session for reference. In addition, it might be better to grade weekly assignments instead of sit-in lab? Or cut down the amount of sit-in labs so students feel less stressed about it. Sit in lab could be implemented still, but grading weekly assignments would be a plus factor.
135. weakness: - unable to analyse code efficiently (take a long time) - unable to come out with code though knows how it works strengths: - able to design algo and come out with pesudo code
136. The module requires students to be able to learn concepts and computing skills very quickly and hence it is difficult for those who lack a programming background to cope with this module. Also the workload is rather heavy as there are labs that account towards the final grading every other week.
137. Very fast paced. Lower the work-load or increase the load for CS1101C to help cope with this module.
138. i think this module is extremely hard for people withput c plus plus background. i feel that the speed of the lecture is very fast and very often than not i cannot catch up with the lecture. as a result i dozed off because i am not following neither i understood it. i believed that the speed is due to the short term and there is nothing much that can be done as the fewer things we are taught also means we are at a disadvantage. i thought that this module shd be like java, where there is part 1 and part 2. maybe this

module can be level 2000 too, cos it is a really challenging module.

139. I think the lab questions are very well set.
140. Would be nicer is the labs are a little more easier. Difficult to catch up in the programming field.
141. Many of us does not have any programming knowledge prior to entering university. And I also observed that though many of us managed to pass CS1101C from last semester, we just merely scrapped through. Hence, many of us are still not very good with programming. With less than a month to brush up our C programming, we are expected to learn C++ programming less than a month too which includes new stuffs like classes and objects and other new syntax and were only being touched briefly. Thus, it makes us even more unclear with programming and starts to deter the module. Although I understands that this is university and we are expected to learn alot of things ourselves, I personally feels that more syntax and coding skills should be covered during lecture as this is not something that we take as normal subjects during our secondary and JC school life. As for stuffs like ADTs, I personally feel that it is easy to understand and what makes it difficult is about transferring the logic into our codes. This is again, because we are not very good with our coding and syntax. Thus, I suggest that the teaching staffs should spend more time one syntax and coding as for ADTs, it is okay to go fast.
142. For lab, i think it's better if suggested solutions are uploaded after the session (both take home and sit in) finished. Lab TA don't give much help.
143. The increase in the diffculty level of the module when compared to 1102c is on a higher side.
144. The module is very interesting to read. However, take home labs should also be graded which will ensure that every student really works to become a proficeint programmer.
145. I felt that this module is one of the modules which is "terrorizing" the students taking it before it actually started.Well I was a bit terrorized.But on the whole I should say that the module was easier than the previous yrs.
146. I feel this module has a very good structure which enhances our ability to develop and design efficient algorithms.
147. important module. develops the basics for other modules and the computers interesting and useful.
148. should be easier.
149. too difficult a module to be handled by students with weaker or no programming backgrounds
150. The professors are very good.The sit in labs were very interesting.
151. strength: enhance our problem solving skill weakness: the module is very tough
152. No comment
153. Need more examples and explanation especially on complexity.
154. none its very good
155. The workload can be reduced.
156. The module stimulates your logical thinking and forces one to apply things learned in class....You teach too many concepts in a very short time. Not enough time to practice and tune your mind to cs1102c type logic
157. the sitting in lab is no good.
158. It is a well designed, comprehensive module! I believe I have learned a lot from it, and I'm sure I'll be using the knowledge in future.
159. This module includes too much content with little time.

160. would prefer to have submissions of lab every week
161. The final should have more weightage(50%) and the sit-in labs should have less weightage(10%) Reducing the course material examinable but still teaching AVL trees was a good idea