

MODULE EVALUATION REPORT

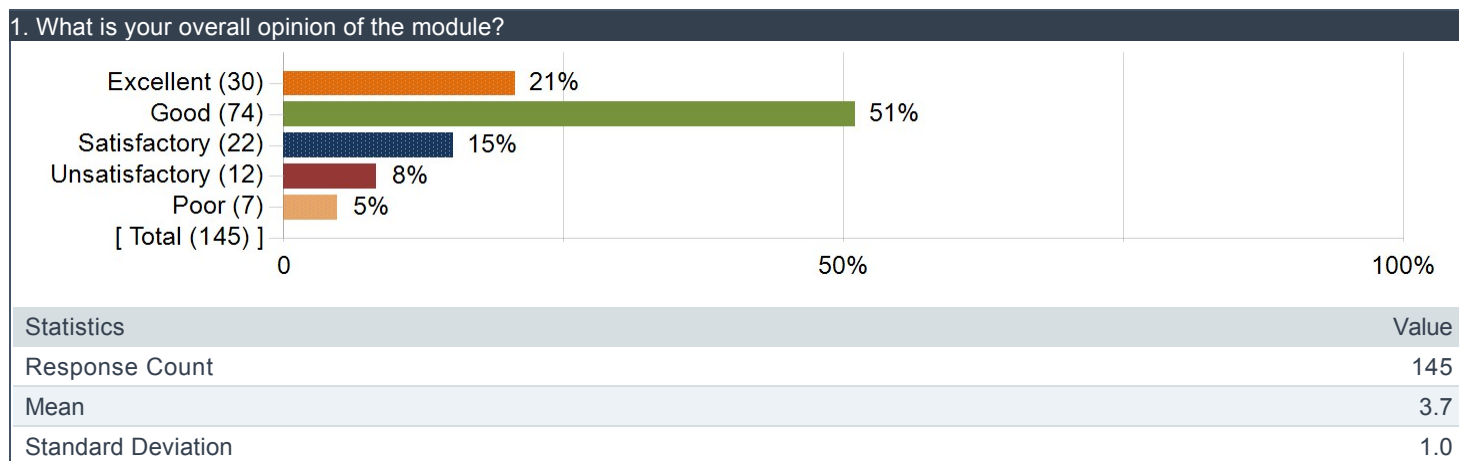
Module	CS1010 - PROGRAMMING METHODOLOGY
Academic Year/Sem	2018/2019 - Sem 1
Department	COMPUTER SCIENCE
Faculty	SCHOOL OF COMPUTING

Note: Class Size = Invited; Response Size = Responded; Response Rate = Response Ratio

Raters	Student
Responded	145
Invited	251
Response Ratio	58%

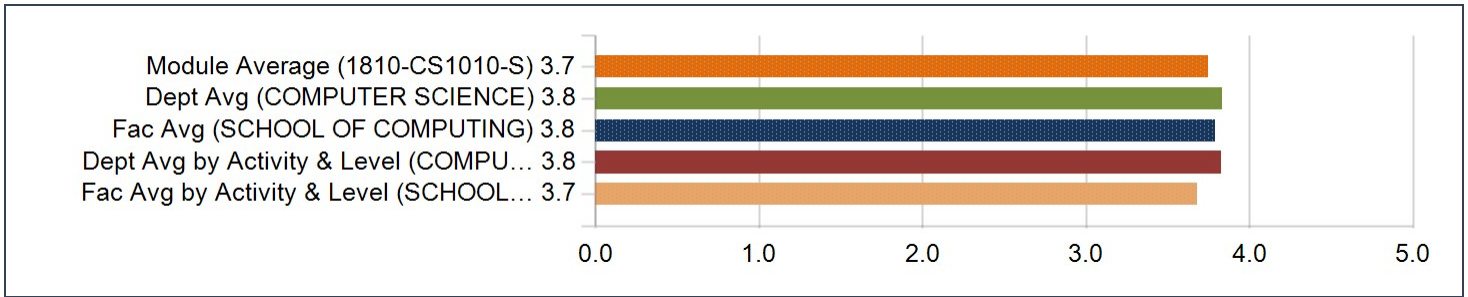
1. Overall opinion of the module

Frequency Analysis



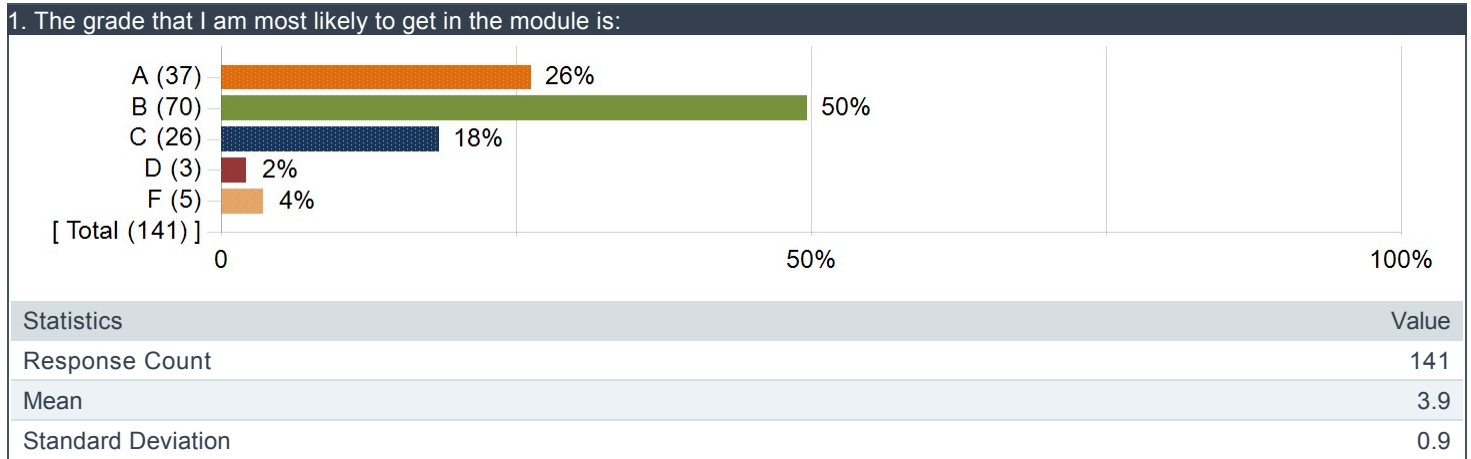
Normative Analysis

Question	Module Average (1810-CS1010-S)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 1000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 1000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
What is your overall opinion of the module?	3.7	1.0	3.8	1.0	3.8	1.0	3.8	1.0	3.7	1.0



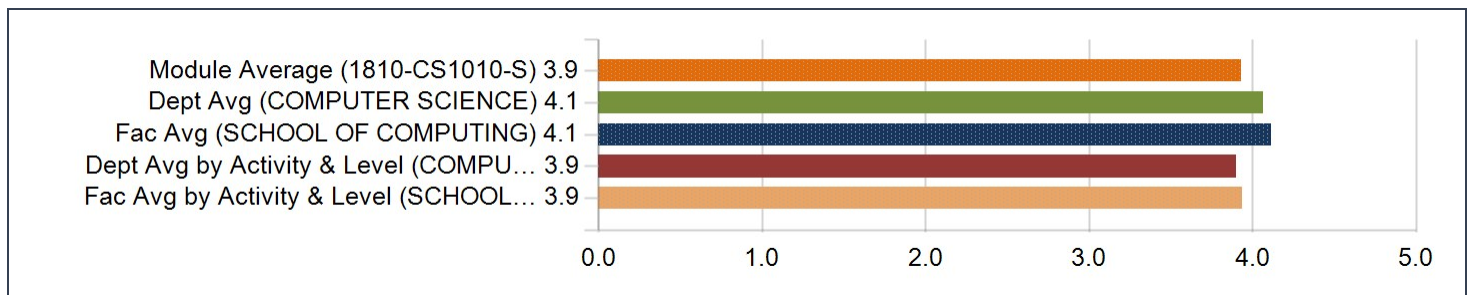
2. Expected Grade

Frequency Analysis



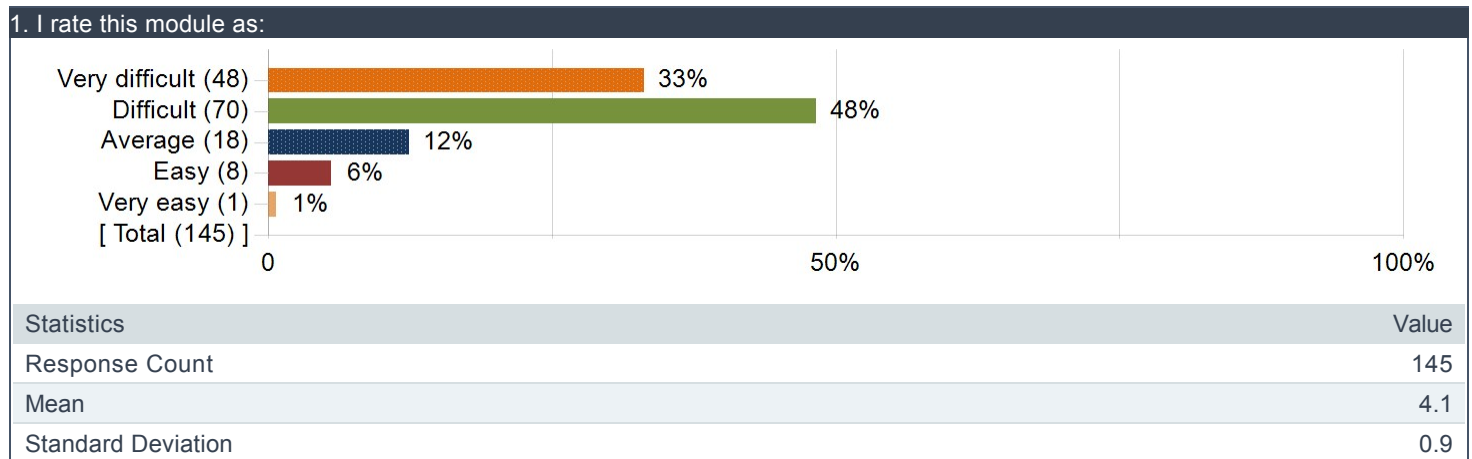
Normative Analysis

Question	Module Average (1810-CS1010-S)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 1000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 1000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
The grade that I am most likely to get in the module is:	3.9	0.9	4.1	0.8	4.1	0.8	3.9	0.9	3.9	0.9



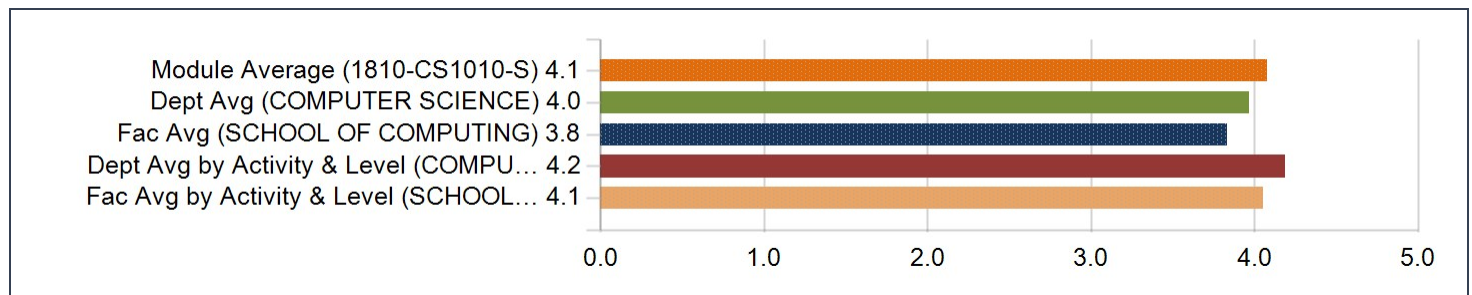
3. Difficulty Level of the module

Frequency Analysis



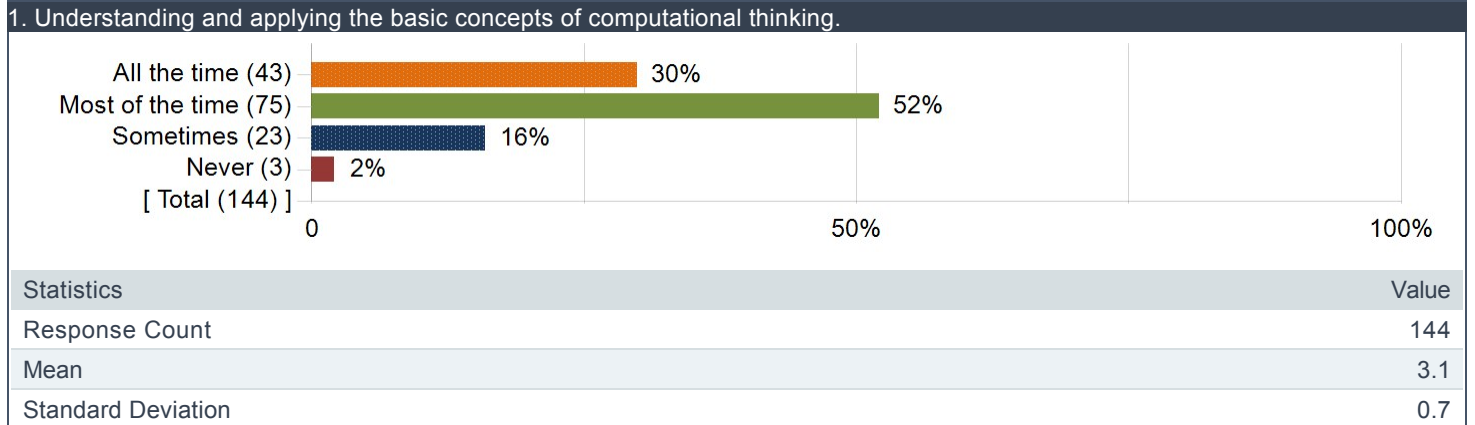
Normative Analysis

Question	Module Average (1810-CS1010-S)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 1000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 1000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
I rate this module as:	4.1	0.9	4.0	0.8	3.8	0.8	4.2	0.8	4.1	0.8

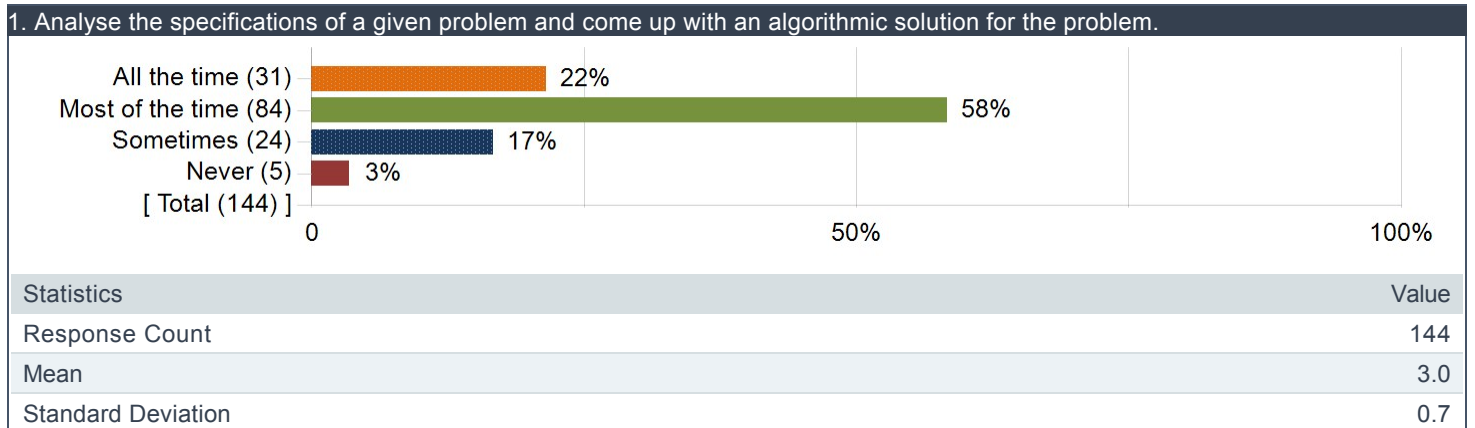


MODULE LEARNING OUTCOMES

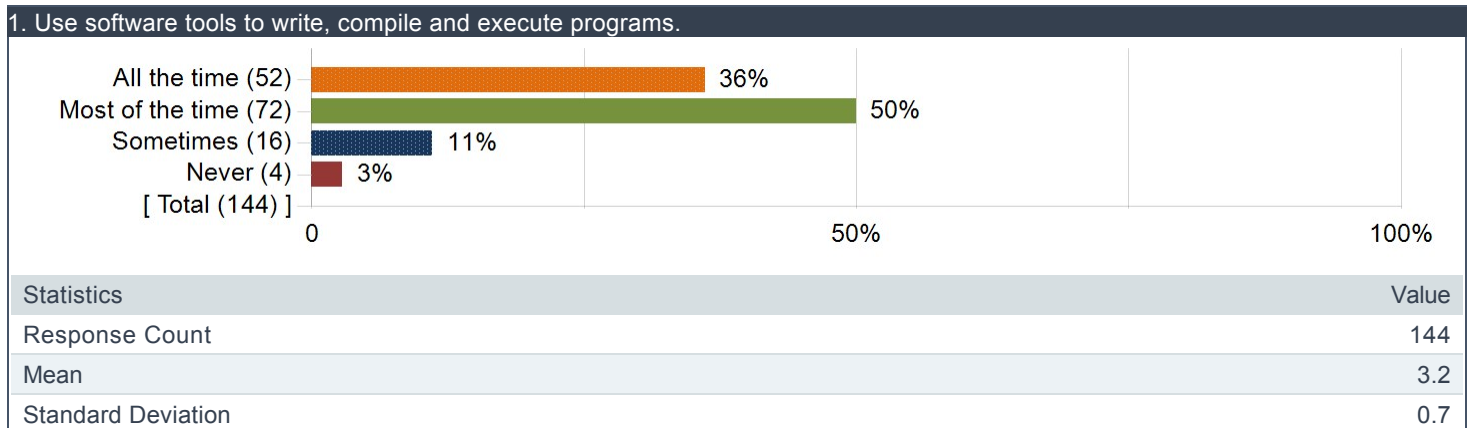
1. Understanding and applying the basic concepts of computational thinking.



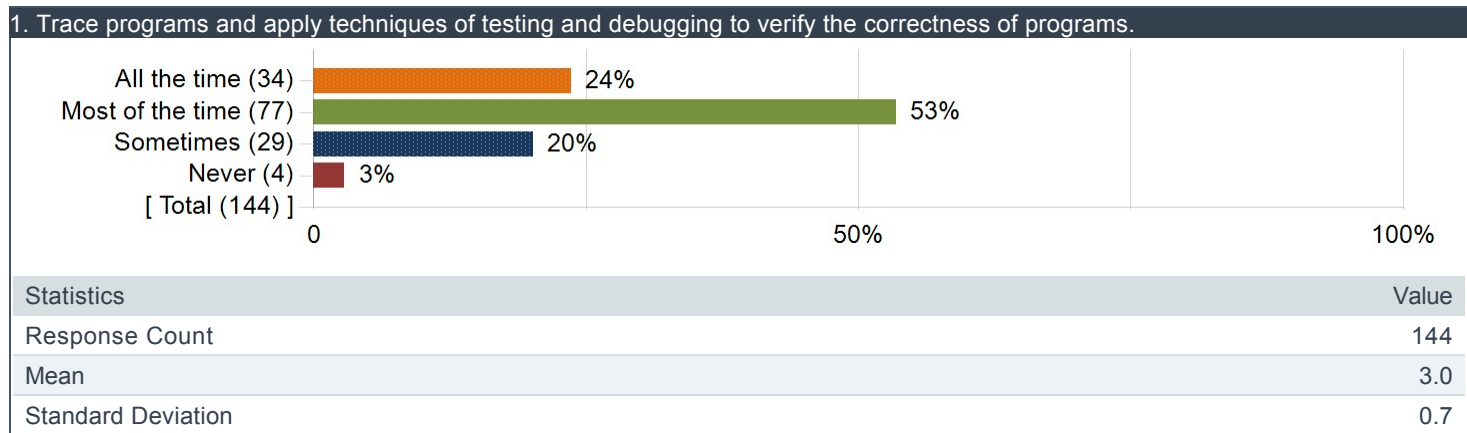
2. Analyse the specifications of a given problem and come up with an algorithmic solution for the problem.



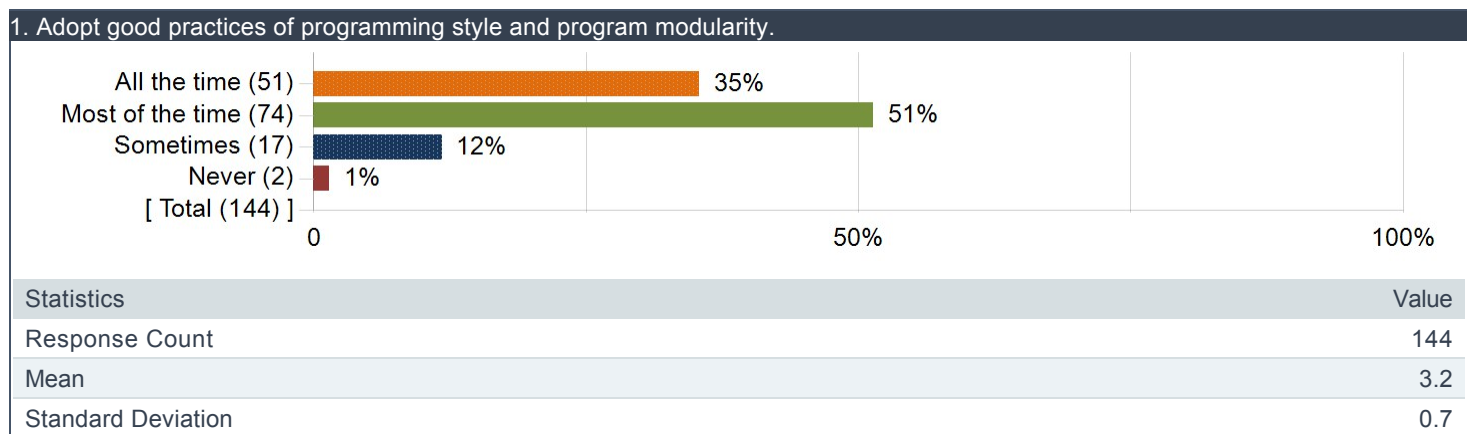
3. Use software tools to write, compile and execute programs.



4. Trace programs and apply techniques of testing and debugging to verify the correctness of programs.



5. Adopt good practices of programming style and program modularity.



WHAT I LIKE / DISLIKE ABOUT THE MODULE

What I liked about the module:

Comments
Even though the weekly assignments had me up all-night at times, I absolutely loved it because it forced me to become proficient and really live the coding life. 10/10 immersion
It starts from the very basics like what is even an operating system and goes step by step. Also, there was always help available from tutors and Dr Ooi.
programming assignments and practical examinations
Good Quality psets. Ease of submission for psets. High quality notes
Interesting programming concepts that form good basis for future modules
Despite the difficulty or tediousness in the assignments, it forces me to read the assignment requirements carefully, take time to plan any codes and improve the efficiency of the codes, which are things that I didn't focus much on in polytechnic. These are good practices that are useful in the future when working on software projects. The structure and flow of the module were effective for me in learning the concepts of C programming.
fun
Interesting concepts and relatively easy for a level 1 module. Assignment questions are also fun to do.
Piazza
-
It helps me learn under pressure

Comments
Something new I never learn before
Interesting introduction to computing and very good foundation
thinking is fun, its always nice to see the 'passed'
the assignments and test questions are very interesting!
Well structured; schedule clearly set-out. Good balance of CA and written tests
-
coding
Pushing us to the limits and forcing to improve.
nil
The teachers efforts to help us
interesting, something different from usual
The assignments were relevant to the topics learnt
Interesting questions and extremely broad scope
Coding
Provide some useful programming strategies and learn more about array and pointer
Coding introduction. Useful for future modules. Improvements to the module did happen after we had a dialogue session with the cohort rep.
Good TA
It challenges one's programming skills despite being an introductory course.
Compared with CS1010 in the past, I like the new changes to it this sem. It's a more 'modern' way of coding in C, and the assignment questions are also carefully designed. I really learnt a lot even though I already had C experience before. Compared with CS1101S, this module has its own focus and strength, which lets me finally understand why this year infosec student must take CS1010. Overall, I will recommend this module to my friends and juniors.
Even though the assignments are difficult, I feel that it has pushed me beyond what I thought I am able to do.
The focus on independent discovery, with guided support
Coding
Interesting and able to learn alot of coding tips and tricks
programming is fun
programming
Excellent teaching paradigm that helped reinforce conceptual understanding of programming, but sufficient discussion of practical examples and issues to enable us to use C effectively.
Develop my ability of keeping working under stress
Challenging and fun. A fresh learning experience. A wide variety of Assignment Questions to stimulate learning.
Teach us how to use c
the questions sets are good
engaging interesting assignments/excercises
Coding is fun!
The interesting assignments.
None.
The assignments
Interesting concepts being taught in some questions
The nature of the subject allows for much creativity.
programming assignments are fun and feel rewarding once completed
Programming is fun
Like programming
taught a lot about the basics of code

Comments
Progressiveness
A sense of satisfaction after submitting assignments
nothing much
None
Learning new things
I learnt many new things that I did not know before such as time complexity of various programs and the programming questions were very challenging and enabled me to critically think and improve my problem solving skills.
Teaches the science behind the programming rather than just how to code (unlike the programming mod from my core modules). I now understand both coding C to a large extent as well as the science behind the codes. Assessment is fair as it is continuous assessment based.
Coding
The materials learnt are very very interesting to me. I feel that I really learn a lot
Introduced to c programming
NIL
Conceptualising solutions to the question was extremely challenging and satisfying
The interesting assignments
interesting introduction to programming
The teacher and interesting content
Learning the basics of using UNIX, VIM, CLANG to write programme in C
even though the weekly assignments took up a lot of time, I felt that they really reinforced my learning, as I am not from SoC and would have very little coding practice outside.
really practical
Interesting to learn a new programming language.
I have always been interested in learning how to code and this module gave me an in depth understanding of c language and how I can use it to solve problems such as maze, tower of Hanoi and mastermind (real life problems).
–
Use of Github for submission was fun
Programming is fun.
Assignments are interesting to do

What I did not like about the module:

Comments
I was assigned the Mac Lab for the Practical Exams and although I admit being adaptable is a requirement for us students, I felt that the unfamiliar keyboard and the disabled number pad made me backtrack a bit more than I was comfortable with given the time constraints.
Heavy workload which was difficult to manage with my other 3k and 4k modules being a year 3 student.
nil
–
Difficulty varies from easy to very hard, too inconsistent with that aspect.
assignments are too difficult for programming intro course. Some of the concepts used (e.g. BFS and DFS) are way too difficult for students without prior programming experience.
Also, the comment and documentation, a tad too time-consuming and useless, for those wishing to enter research area rather than private industries
I think there should be some exercises to build our skills and then move on to the assignment. As it stands, the lecture is like teaching $1 + 1 = 2$, then the assignment is like if $1 + 1 = 2$, using this information, calculate the mass of the sun. There is a big gap in knowledge between the lectures and weekly assignment.
–
This module has heavy workload but the time for revising old lectures is limited

Comments
Too fast for me understand
weekly workload is too much and a lot of trial and error is needed to complete assignments. This mod is very time consuming but it is not fun when I don't have enough time to think.
high workload for level 1000 module
The weekly assignments were tedious, but I understand that programming is a skill that needs practice, and that this will bring benefits later on. In addition, the learning curve is quite steep early in the course for someone who has no coding experience (What's IDE, Vim, etc.?)
Workload too high
coding too much
No positive feedback when one improves. No breaks at all, deadlines every week. Too time-consuming and mentally taxing for the weekly assignments.
PEs
Pretty much everything, from the choice of the programming language to the methods of teaching. The teacher almost kind of assumes that this is the only subject we do.
workload very heavy!!! i didnt like how the assignments were so much more difficult than what was taught. maybe can smoothen out the difficulty levels of the questions bc i feel like this mod has a very steep learning curve.
The frequency of the assignments.
Module not suitable for ppl without prior programming experience.
Written paper, mcq too high weightage
Too many assignments which need a long time to finish every week. Really stressful.
Second PE lack time
Had to really rely on friends who knew what they were doing beforehand, and it is really difficult to learn on your own when every week is a deadline.
Overly strict documentation rules Forced to toggle between three sites for various things (Piazza: forum, github: Notes/Assignments, IVLE: Gradebook) Not even willing to give TAs the flexibility to award marks for partial completion/ideas present/concepts displayed
The pacing of the assignments and the level of difficulty can sometimes be hard to cope with.
maybe feedback can be given to us more quickly (like feedback on PE1)
I feel that my theory regarding this module is weaker as compared to my practical knowledge. Lectures are too dry for me to follow.
The workload at times was very demanding
Difficult to catch up with people who already have prior experience CS1010 I/O can sometimes feel like a crutch to use
Weekly assignments are taking too much time which affects the amount of time spent on other modules
some questions are damned difficult but still fun nonetheless
programming
Did not cover certain technical aspects that would be considered quite important, such as file processing and string functions.
Stressful
Just Assignment 5, there was no way anyone would have thought of using matrix multiplication for that one.
we have to use vim
Learning C.
the time given for each assignments
The weekly assignment and very challenging and takes too much time while i think exercises that test on concepts and basics are not enough.
The hard assignments.
Too strict and there are too much limitations on what we can/cannot do. Stuffs learnt whilst the most part is introductory , the assignments are very abstract and i did not have time to fully absorb the complete meaning behind the fundamentals learnt. I would rather have more assignment questions in the form of drills and concepts we have learnt (basics), followed by maybe one average-hard application question. i dont feel like i learnt a whole lot, just felt like i was vomiting out what i learnt the previous week
Too fast paced;
From the view of someone who is a complete beginner to coding, this module was not welcoming at all. Though our CEG level

Comments

representative took a step forward to suggest coming up with extra classes for those in need of help and to lower the assignment difficulty levels, none of it was implemented. This module to me was just never ending incoming waves and in order to conquer them, I guess I'll have to pick up on my own during winter break.

Too fast pace and too much content.
Assignments given too often and too difficult to be completed.
Demoralising.

NIL

Weekly assignments are a terrible idea. Even the TAs can't grade the assignments on time. Teaching efficiency concepts is fine, but it doesn't seem like an introductory course material.

Quite difficult in the beginning if student has no prior experience.

no idea how to prepare for written exams which make up most of one's grade other than to revise the notes

Workload a bit heavy, not enough preparatory materials/advice for major exams. PE too rushed, not conducive for learning. May put off the less skilled students, leading to lack of confidence in future mods involving programming

the assignments took up 90% of my time and i had very little time to study other mods

Abstract and hard to follow on some topics, such as recursive backtracking

Don't understand most of the topics taught, I have to constantly self-learn to understand them

too many assignments. it wasnt a fun experience learning C programming

This module is really very difficult and unrewarding. Partly because of the professor but also because programming is made unnecessarily difficult since it's not really required to such a degree.

The ridiculous difficulty level

The learning curve was quite steep for students with no prior programming experience

Workload relatively heavy

Abstract things like heap

The assignments are very tough.

TOO Fast paced, assignments were very hard and not at all building up from basics to hard materials. Felt like students needed at least prior knowledge of coding and was not "introduction to new learners". Students who had absolutely no prior knowledge to coding had hard time catching up.

The module started off easy to follow and at a level where students with no coding experience can follow the class. However, the workload, the weekly assignments and the way the module was structured was poorly designed and was not designed to teach the "no coding background" students. While the use of flowcharts did help slightly, many times the prof would type the code on the screen and compile without much visualisation. It is hard to follow when the code just runs without allow the students to follow how the code runs like on visualisation app such as on python tutor.

The weekly assignments were not just a chore to do but the difficulty level was not properly moderated to the average student. Many times students could follow the code logic of their stronger peers or seniors and heavily alter it to prevent the plagiarism checker. This defeats the purpose of learning and doing the assignments and disadvantages many students. What could have been done better was to split the exercises and assignments to not coincide in the same week. More can be done during tutorials was such that the assignment could be done and submitted during assignments, instead of multiple questions, there could be one-timed assignment each week on the topic. Exercise would be done at home instead. This way it would be better to gauge the students understanding of the subject rather and his ability to execute the code from the lecture. Having the graded weekly assignments coincide with the ungraded exercise would ensure the full attention of the student on the assignment and not on the exercise.

In terms of the module's objective, having efficiency included in many graded assignments was not effective at teaching them how to make their code more efficient, instead, it punishes students especially the weaker one heavily. In assignment 7, 0 marks are awarded for solutions below $O(n^2)$ timing. Having such grading criteria would not help the student from learning code but rather forces them to find a "quicker" solution which they not understand. Furthermore, their quicker solution may result in segfault which in turn gets 0 no matter the circumstances. This form of grading is not helpful in learning code at a beginner level. Instead of giving 0 when the code is not able to compile, marks can be given if the logic is acceptable and capped at a certain percentage, this method would allow the students to learn and understand coding logic better.

To further add on, while the no bell curve was one way to make the mod manageable, I strongly feel that this mod can be made into a pass/fail module instead of graded from A to F with its current difficulty.

Problems with the language which resulted from an inability of the code to read what I intended for it to read

The weekly assignments.

Comments

The examination and practical paper are not meant to test understanding. What kind of paper is set so the average percentile only completes less than 50% of marks?

assumes students all have programming experience/has a very steep learning curve at the beginning compared to other programming classes i have tried. feedback on PE1 is very slow

The weekly assignments

The weekly graded assignment is taking up too much time to complete in order to meet the deadline. The difficulty of assignments jump rapidly from the previous week without practices on the new concepts taught.

perhaps the university can look into separating students with coding experience to better cater to all students

really hard for students with zero prior knowledge to catch up with weekly assignments

The weekly graded assignments posed a heavy workload and it was very difficult to balance between them and other modules.

The pace of teaching may be a little fast as I'm a beginner with no prior knowledge. There are many times I would have to Google to find out more as the prof explanation may not be clear at times.

the steep learning curve and lack of time to improve

Hard to do up the coding sometimes.

The PE2 was incredibly difficult and not enough time and practise was given to the students that will help allow us to answer the questions in the exam. Also lack of documentation for the library that we are using in our code. One line of code(which was quite hidden) in a website cost me a lot of marks.

Tutorial time is not well utilised