



## QMUL Module Evaluation Scheme: 2023-24 SEM 2

Dear Thomas Prellberg,

This email contains evaluation results for Introduction to Computer Programming / MTH5001/B23/SEM2

The results for the fourteen QMUL core statements are listed first, followed by those for any school-specific questions.

To provide a visual aid to help differentiate between more and less positive results, traffic light 'quality indicators' have been incorporated into the report. These are marked according to the mean score for the fourteen statements, scores of less than 2.00 will be marked red, scores of 2.01 and 2.99 amber, scores of 3.00 and above green. The median score is also included for each statement in the report.

The scale on which students scored their views is as follows:

- 4 = Definitely Agree
- 3 = Agree
- 2 = Disagree
- 1 = Definitely Disagree

Schools and institutes are encouraged to use the traffic light system above and information from other sources such as Student Staff Liaison Committees to monitor module performance.

Also included in the report are the free text comments.

We hope you find this report useful, please do continue to provide feedback on the evaluation process to your School Managers who will pass it on to the Module Evaluation team to enhance the scheme.

Thank you.

The Module Evaluation Team: [module-evaluation@qmul.ac.uk](mailto:module-evaluation@qmul.ac.uk)

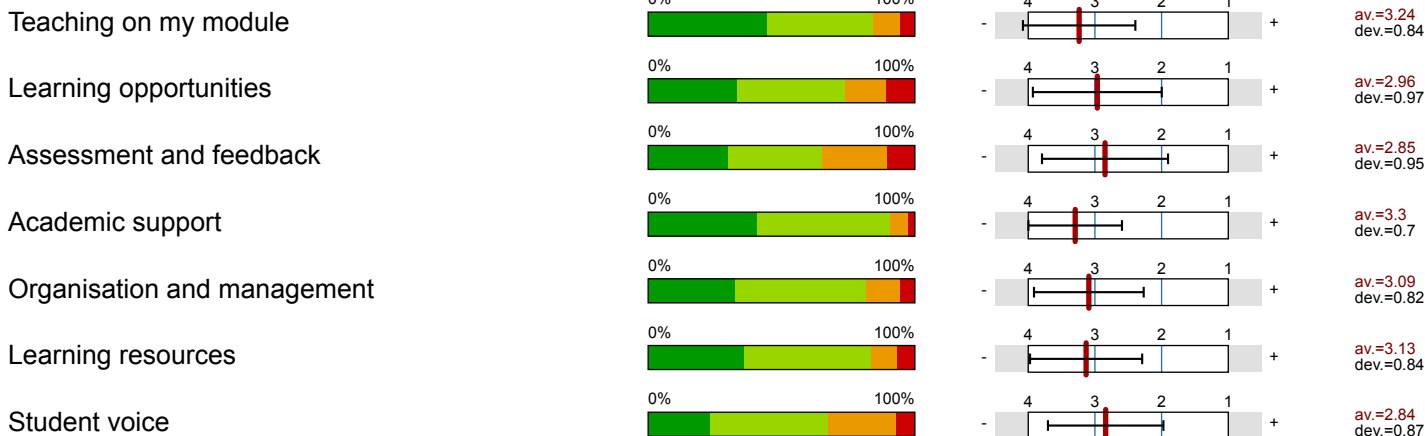
Thomas Prellberg

Introduction to Computer Programming (MTH5001/B23/SEM2)

No. of responses = 56 (14.4%)



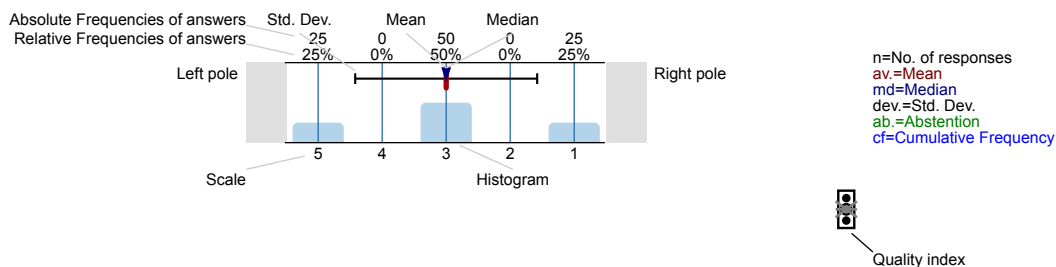
Overall indicators



Survey Results

Legend

Question text



Description of quality symbol

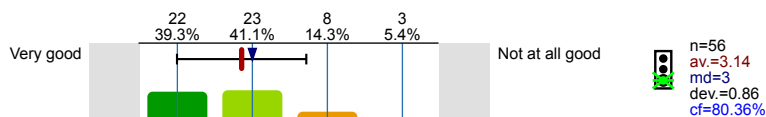
Mean value is below the quality guideline.

Mean is within the range of tolerance for the quality guideline.

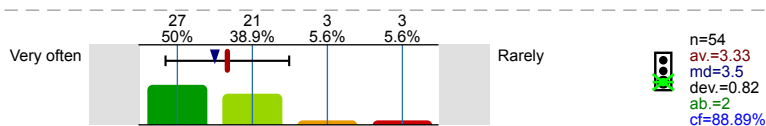
Mean value is within the quality guideline.

1. Teaching on my module

1.1) How good are staff at explaining things?

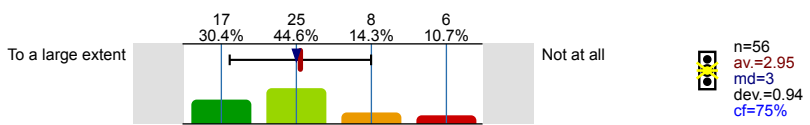


1.2) How often does your module challenge you to achieve your best work?

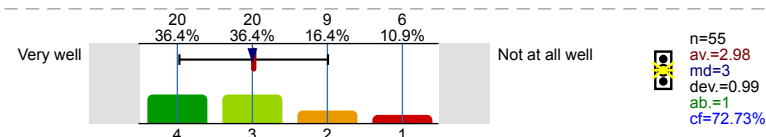


2. Learning opportunities

2.1) To what extent have you had the chance to explore ideas and concepts in depth?

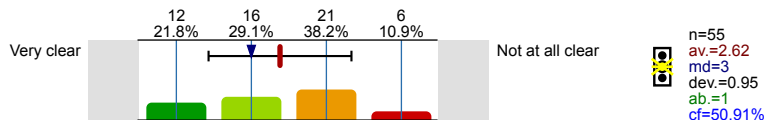


2.2) How well has your module developed your knowledge and skills that you think you will need for your future?

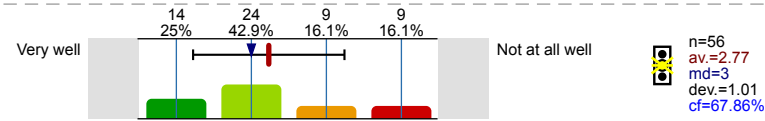


### 3. Assessment and feedback

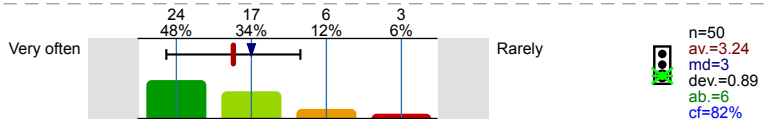
3.1) How clear were the criteria used for marking your work?



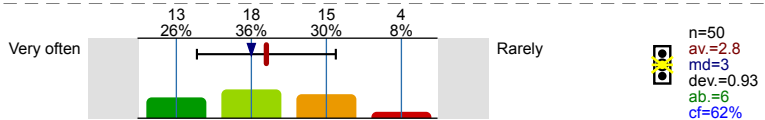
3.2) How well have assessments allowed you to demonstrate what you have learned?



3.3) How often have you received assessment feedback on time?

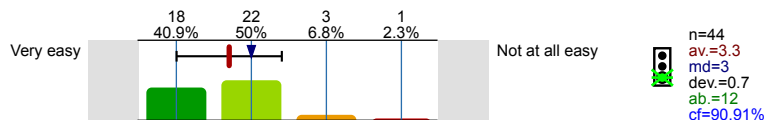


3.4) How often does feedback help you to improve your work?



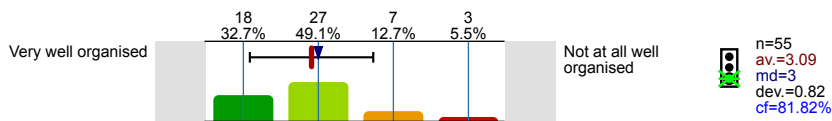
### 4. Academic support

4.1) How easy was it to contact teaching staff when you needed to?



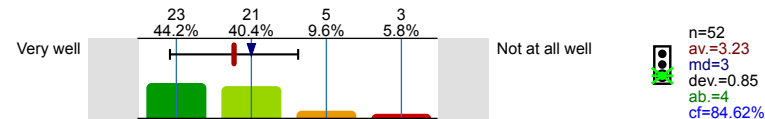
### 5. Organisation and management

5.1) How well organised is your module?

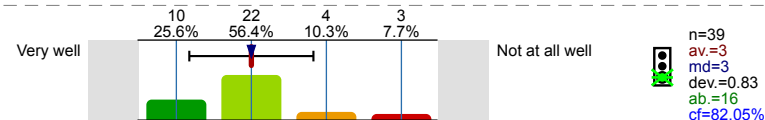


### 6. Learning resources

6.1) How well have the IT resources and facilities supported your learning?

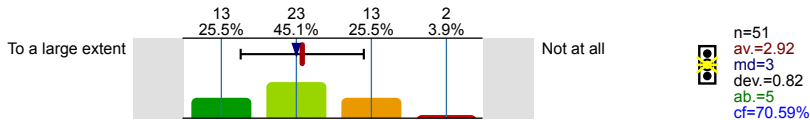


6.2) How well have the library resources (e.g., books, online services and learning spaces) supported your learning?

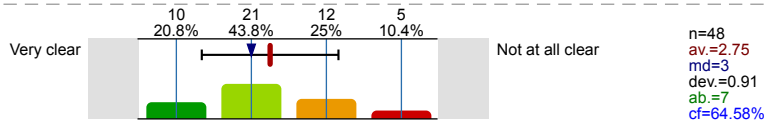


### 7. Student voice

7.1) To what extent are students' opinions about the module valued by staff?



7.2) How clear is it that students' feedback on the module is acted on?

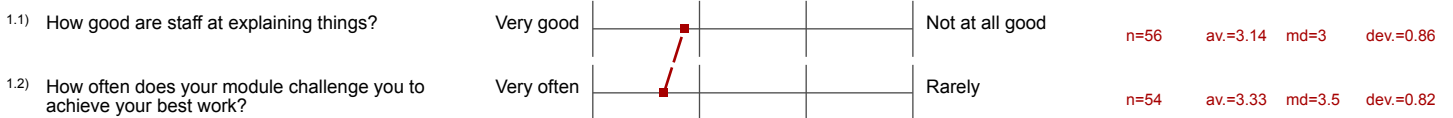


# Profile

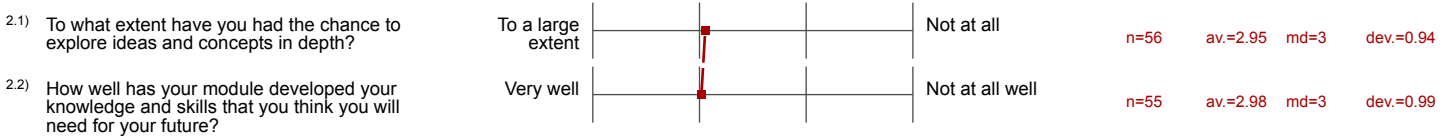
Subunit: School of Mathematical Sciences  
 Name of the instructor: Thomas Prellberg  
 Name of the course: Introduction to Computer Programming  
 (Name of the survey)

Values used in the profile line: Mean

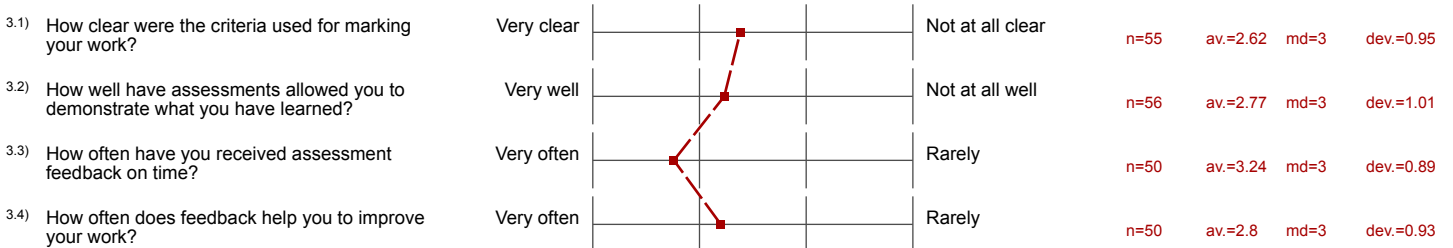
## 1. Teaching on my module



## 2. Learning opportunities



## 3. Assessment and feedback



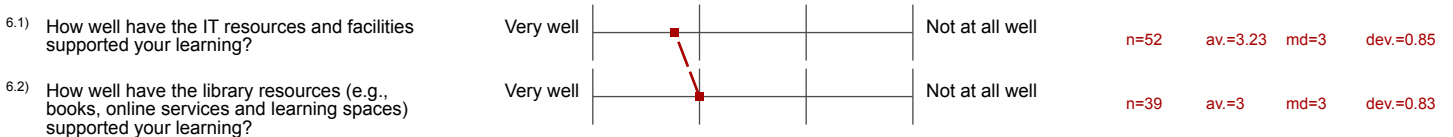
## 4. Academic support



## 5. Organisation and management

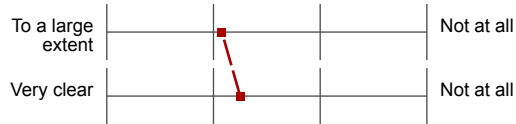


## 6. Learning resources



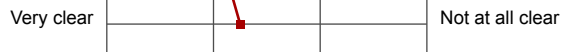
7. Student voice

7.1) To what extent are students' opinions about the module valued by staff?



n=51 av.=2.92 md=3 dev.=0.82

7.2) How clear is it that students' feedback on the module is acted on?



n=48 av.=2.75 md=3 dev.=0.91

# Comments Report

## 8. Your comments

8.1) What are the best things about the module?

- being taught the basic understanding of python and what coding is.
  - I am not familiar with Python, but both lecturers have taught often complex content well, which has allowed me to complete the lab sheets well and to the best of my ability.
  - Interesting and practical to learn. The labs and lecturers are very good at teaching me what approach to take and are straight forward to follow.
  - Interesting and thorough module. Challenging. Develops skills for future careers
  - It's fun to learn python and its applications and both Thomas and Lennart are very enthusiastic and good at explaining things.
  - It has a lot of interesting content. The lectures and labs are easy to follow as a complete coding beginner. The mix of math formulas in the tasks is good.
  - i think the weekly quizzes and lab sessions really help me hold information longer also helps me to use information I learn in lectures.
  - It is very interesting and well paced
  - Lab sessions are useful  
Great walk through as well
  - Learning python
  - Lecturers going through the lecture notes on Tuesday and Friday.  
Help in lab sessions.
  - n/a
  - nothing
  - Nothing.
  - Nothing to be honest !
  - Taught in a fun and engaging way. Organisers are fun and helpful in labs. Actual content is mostly relevant and helpful in the real world. Lab walk throughs are helpful and a great addition.
  - The content taught is good and it makes sure that you learn python properly. If people don't turn up, it's their fault. Nothing can be done.
  - The format of assessments being two in term exams in weeks 7 and 12 worth 10% each and a final project set over a few weeks worth 80%.
  - The IT labs are very useful
  - the labs and the walkthroughs
  - The labs are very helpful for learning
  - The lecturer
  - The lecturer is so nice!!!
  - The lecturers clearly know what they're doing, they're enthusiastic, more than other lecturers I have had before personally. They explain in great depth when necessary and have a good sense of humour. They provide all necessary resources such as the walkthroughs which are incredibly helpful to revise and learn why and when certain pieces of code or comments are needed. The walkthroughs are also very helpful to people who cant make it to the labs so I very much appreciate it, thank you!
  - The lecturers who make it very fun and engaging. Also the labs are great. Overall very good module.
  - The lectures are incredibly in depth and go through things very well
  - The lectures are lovely and amazing. Especially Dr Thomas , very helpful.
  - The work is fun and a lot of examples are either gathered from mathematical concepts or are applicable to areas outside computer programming.
  - This module has a more realistic outtake on Mathematics. In the real world I will have access to internet, calculators, and different online tools so making the exams and courseworks open book with access to these tools are extremely beneficial as i can learn in a more modern style of 2024, rather than old school non-calc, closed book exams which will never apply to me in the real world.
- Staff were kind enough to provide model solutions for the labworksheets provided which i beleive was a great way to learn also since i

can see how an expert can tackle certain questions.

- We're learning something useful and it's examined with a project rather than an exam.
- We have many resources to practice, like quizzes and labs
- We learn how to program. Yay
- Well organised  
Well taught, concepts explained well and if not understood the teaching staff are good at helping you to understand the more complex ideas.

8.2) In what ways could the module be improved?

- Could release the final project earlier so we know what it will be able and will have more time to prepare for it before the other module exams start
- do force people to do the module, just abolish the whole module
- Explain the answers to the assessment questions most failed in depth
- I do not think this module should be compulsory. The university assumes all Mathematics students "need" to know Python for our futures when in reality the majority of us do not for our professions. It is unnecessary to make every student do this as a compulsory module and should be elective instead so I can choose to learn other modules that do apply and interest me. I also think students access could be improved, a large number of students in mathematics use tablets or paper notes over laptops due to mathematics being a much easier subject to write by hand. The lecture notes for this subject were all in a Jupyter Notebook form that was difficult to access for these students not using laptops. In the first week the lecturers converted the notes to a pdf which was much easier for everyone but then refused to do this again for any following week. This meant that myself and many other students had to come into university and into the library to work on the computers to be able to access any lectures to make notes and revise. This was a massive inconvenience especially for students who commute and live quite far away. Not having easy access to revision and learning resources was a big obstacle in my learning this module which I definitely think could be improved on for future students as tablets become much more popular especially for students in this subject.
- I feel like the module did not really teach python properly and just threw you at problems especially for people who have never done python. Find it very difficult to do questions, worried about the module as a whole.
- I feel that the examples in lectures could potentially be harder to match the difficulty of the labs and the interim assessment and quizzes, however I understand this is very hard to do as there are many people to accommodate.
- I find it slightly odd that im doing a python course in a mathematics course. I understand this is the new trend, however for certain questions i may get correct, i may receive the incorrect answer due to a simple syntax or coding error despite me getting the general method and approach correct. This is rather tedious and i believe doesnt really test the mathematical aspect of my knowledge rather the coding side of my knowledge, after all this is not a computer science course that i payed for!
- improvement in the labs to explain any confusion on parts of the work (e.g. hinting where we went wrong instead of just telling us to read it again "closely").
- It could be more difficult with more contents
- Jupyter notebooks can be bit cumbersome to use on appsanywhere - would prefer pdfs off the bat even if they are easily converted at home.
- Lectures explain the very basics of Python and this is troublesome as the quizzes and coursework expect you to have detailed and in-depth knowledge of Python, displayed through the questions, which expect you to write detailed code. I cannot apply what I learn in lectures to the questions in the coursework and the quizzes as these questions are levels above in difficulty compared what's explained in lectures. Even in the Labs, the questions are astronomically harder than what's explained in lectures so it's difficult to go through the sessions without a vast amount of help and some sort of independent thinking.
- Lessons could be more interactive
- more in-depth explanation
- More IT lab sessions instead of lectures
- Not much I can think of
- People who has/with no knowledges of python to learn these contents is very ridiculous/ impossible (they really need to consider this) and this module is called "introduction" to programming and this is the worst module ever! The professor is a bit arrogant! Bad designed materials! The convenors don't know how to make and organise the materials like the lecture notes, exercises and quizzes and lastly the questions in the quizzes are very vague!
- Quizzes don't reflect what's been learnt in class, same with exam. Not every staff member is easy to communicate with. Mid term exams
- the appsanywhere is terrible it crashes so often on both my laptop and the uni library computers
- The assessed coursework included methods of working out solutions that were not similar to IT classes. Under a 3 hour time pressure it was very overwhelming.  
For example, coding using if and else statement was alright but was expected to make changes to elements that were not tuples of lists, which is what we have mainly been using in labs
- The feedback from past student surveys does not seem to be taken into account
- the format of the course work was a little bit harder than i would like
- The hidden tests in the quizzes are awful. My code will work, yet I receive 2/10 and when I look at solutions, the only difference is one line of code. This needs to be addressed for the Week 12 assessment and future years as it is just not fair.
- The quizzes, particularly Coursework 1 felt very far removed from the content we had been studying or the questions in the lab sheets. I did feel that the questions were far harder than any other questions we have encountered, which does make one feel slightly dejected, and has made me worry as to how difficult the final project will be.

I would be greatly appreciated if the lecturers could do a walkthrough of Coursework 1 (in person or on video) and explain the reasoning behind the solutions to each questions.



I am not familiar with programming, and I know many others are not either, and I do feel this module seems to cater more for people who are familiar with programming, particularly Python.

I understand that programming cannot be rote learnt and there can be several ways to reach the same answer, but with some of the coursework questions I often didn't even know where to start, making certain questions inaccessible, and it does appear that some methods, even if they are correct are not accepted, which again is at times frustrating. I do understand that the lecturers cannot foresee every possible variation of code, but it is, as mentioned above, a cause for concern for the final project.

- The quizzes are too difficult  
Maybe there could be notes on how to do those
- The two hour time limit on the in term assessment was too rushed. I felt like I had no time to think and that result does not reflect my best effort. In addition, while I understand that the code doesn't work if there are small punctuation mistakes, I feel like some credit should be awarded if there is evidence that the question has been understood.
- The weekly quizzes and lab sheets are very difficult. They do not seem to match the level of examples shown in the lectures. The questions are very thorough and very complicated. After week 5, it became rare to actually be able to answer a question on the weekly quizzes. Sometimes, the code may work in Jupyter notebook but fails the hidden tests on qmplus. Also, it is quite difficult to follow along in the lectures when we are writing notes due to the speed of the lectures. It would be a good idea to have these lectures in a computer room so that we can follow along with the Jupyter notebook lecture notes and practice the code ourselves.
- This module is not for me !!!! (many such cases). This is the most basic module I have encountered.  
  
Maybe because for this reason, people do not turn up.  
  
I think it should be more independent work and labs, because you can't really learn from a lecture.
- To extend the lectures on the basics of python for those who have never coded before. To also extend the time taken for in-term assessments as thinking and typing time is variable for everyone. To reorganise the learning structure as the lectures themselves are quite hard to follow without a computer setup available to practice what we learn directly.
- – It always seems like the lectures, the labs and the in terms assessment are all very different things. It needs to be remembered that this is the first time a lot of us are learning python so you can't just give us the basics/ the bare minimum on lecture and the make the tutorial be at a strong python user's level and the in terms assessment at the level of the lecturers. Please work on that  
– For the Lecturer – when asked questions in the labs, please try not to speak down to us because we struggle to understand and grasp something that is completely new to us. Just be patient and explain it properly. And also don't assume we know PhD level maths or anything like that, we only know what we have learnt and so when you say “as mathematicians, you should know (outrageous maths) that we don't know because we have never been specifically taught it, it feels really bad.

Thank you