# PX150: Physics Programming Workshop (20/21)

## Feedback on module

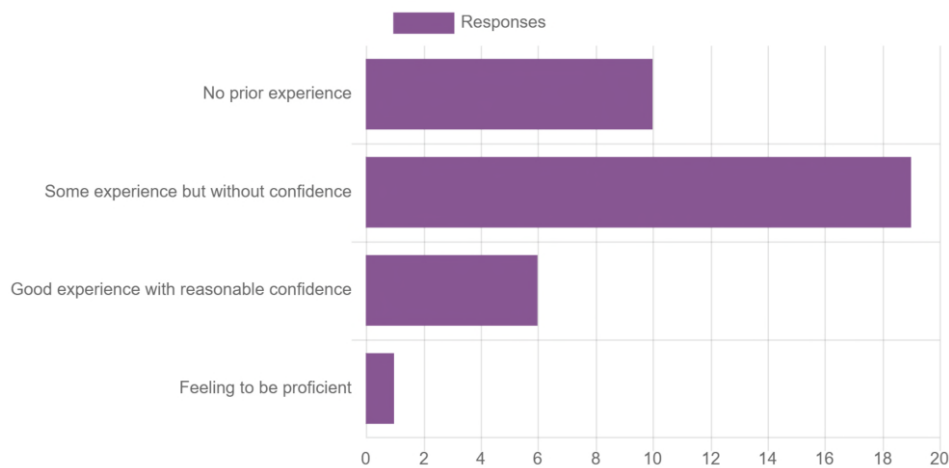Overview    Edit questions    Templates    Analysis    Show responses

Visible groups: All participants
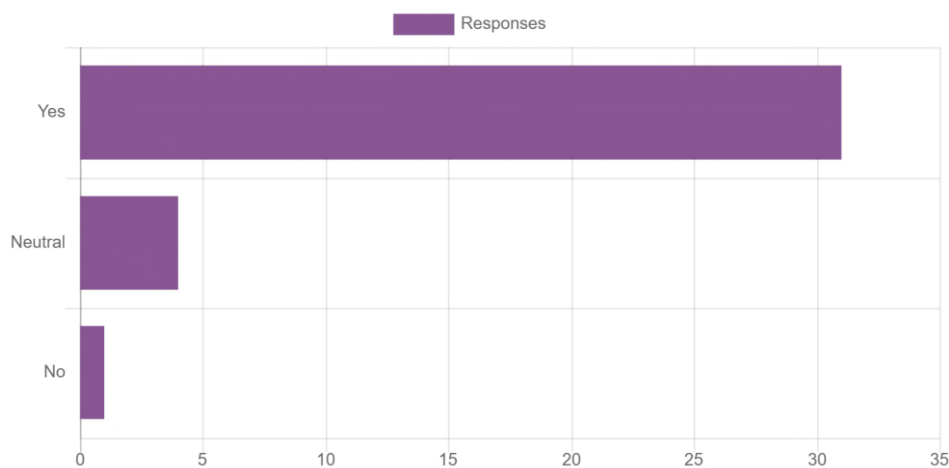
Export to Excel

**Submitted answers:** 36 / **239**
**Questions:** 8
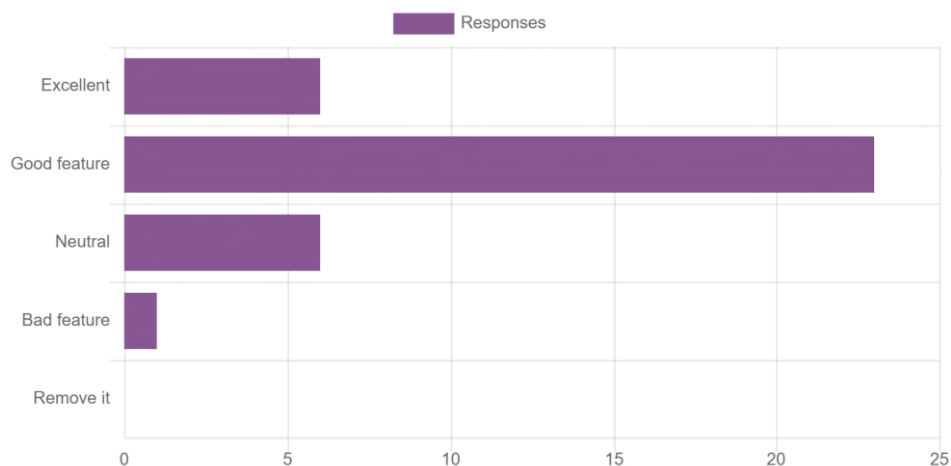
**What was your prior programming experience with python**



Show chart data
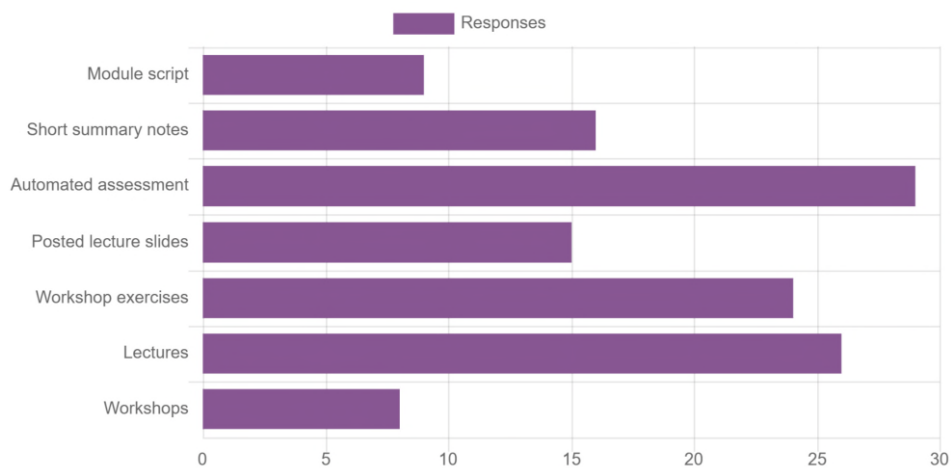
**Would you like a module taking this subject further?**



Show chart data

**How would you rate the automated assessment process on moodleX**

| | Responses |
|---|---|
| Excellent | 6 |
| Good feature | 23 |
| Neutral | 6 |
| Bad feature | 1 |
| Remove it | 0 |

Show chart data

---

**Which feature(s) of this module did you find valuable or useful? Multiple answers are permitted.**



| | Responses |
|---|---|
| Module script | 9 |
| Short summary notes | 16 |
| Automated assessment | 29 |
| Posted lecture slides | 15 |
| Workshop exercises | 24 |
| Lectures | 26 |
| Workshops | 8 |

Show chart data

---

**(Pro) The best features of this module were:**

- The example codes given as they were very useful to refer back to an adjust to fit the requirements of the code needed

- Writing the code, helped get an understanding. Short lectures were refreshing.

- The workshop exercises

- The lectures and the workshops

- Good Exercises

- Having a module with practical elements where my problem solving skills are used rather than just video watching and note taking

- I think the style of the assessed question was excellent as they asked more physics-based problems etc.

- The lectures and accompanying slides

- Plenty of information on how to get our own IDE and run code ourselves, Easy to find notes and lectures, I love michal.

- The variety of problems in the assessment/exercises. It made sure that everything we learned that week was used as well as forcing us to research extra information and get used to the python documentation.

- Using the Moodle assessment helped to give me a grasp of making my own code, practising this is the most important thing IMO.

- Lectures (I revisited and paused the videos rather than using the notes).
Spyder recommendation (it was much nicer to code in than the default Python interface).
Automated assessment was preferable as, even when mistakes were made, I could address them and still get most of the mark.
The test console is a good idea to avoid losing marks.

- The workshop and the assessment were really good at making you focus on the code and trying different methods

- Lectures, workshop

- the amount of live sessions and help available

- I think the videos were very good and the example code was very helpful when going on to do the assessment problems.

- Definitely the workshop exercises, really useful to practise. The notes helped a lot too.

- The independence

- instant feedback on moodelx

- Concise lectures

- workshop exercises, harder than assessment so good for practise

- The lecturer explains things in a concise clear manner. The code notes are very useful for solving the XMoodle Assessment problems The section on numerical analysis using Numpy and SciPy was my favourite set of lecturers. A lot of previous python courses I have attended came from a computer science angle, not really making much use of python modules, instead focusing more on learning object orientated programming. Being able to make use of Numpy and SciPy gives python a real application for me. The fact the XMoodle problems were physics related gave a meaningful application to what I was learning. This course starts from the basics and gives a good solid foundation for future programming courses. It has made me consider future modules such as computational physics.

- The exercises help us learn and understand really fast individual Python functionalities before engaging with the assessment.

- The assessments and workshop exercises, as they highlight a very broad range of areas where programming can be extremely useful.

- The module script was very useful. Lecture videos were a good introduction to the week's content but my main source of knowledge was the module script.

---

**(Con) Any particular aspects needing improvement (and suggestions how):**

- When needing to do own research for a certain code (i.e codes involving numpy or math), some guidance on when we need to do that research would be helpful as sometimes you write many lines worth of code when there already exists a function that can do it in one line.

- It would helpful if moodleX could tell us what specifically is wrong with the code because sometimes we don't know why our program is wrong and what tests did moodleX use to check our code.

- Maybe a more interesting task at the end of the module that would bring together what we've learnt throughout.

- It could be helpful to have a few worked example problems to look over alongside the non-assessed problems so that those of us with little to no prior experience can look at some more examples and have a better idea of how to do the simple things.

- Sometimes it was hard to see where errors were in a code as the the error checker gave weird outputs.

- Some of the slides were worded slightly weird, but they were still very useful

- More information could be provided on each week's topic on that weeks lectures or lecture notes. This can help because it will be a reliable, concise and an easy to access source, which would be better than browsing through many sources on the internet to find something that is reliable.

- For people with no previous experience with Python, a lot of Googling had to be done for the assessments - some more useful tips could be included in the lectures/slides e.g. how to read certain columns in a file, how to go through all variables in a range

- Possibly a sheet with all the basic python commands and how to use them
If there is a way for MoodleX to give more detailed feedback when code doesn't work that would be handy

- Test python console could give the output(s) of the code

- Some of the questions were a bit hard to interpret because they were based on unfamiliar physics e.g. the Lorentz transform matrices.

- Lectures needed work, the format of watching someone code doesnt work very well in practise. Maybe write more detailed notes to make up for.

- no

- N/A

- Sometimes instructions for assessment weren't clear or specific, which was frustrating. Lectures and notes needed to be clearer with more examples so applying principles could be easier.

- More information on the various topics, or atleast more direction. Spent long periods of time, conducting research, which was at time exhausting, as I felt miss-directed.

- When we need to check if the input is less than $10^6$, mention whether its inclusive or exclusive

- nope

- advice on how to use the Moodle console to avoid minor errors

- Some of the questions were a bit vague, it wasn't clear exactly what the task was and nor was it clear as to what moodleX was marking.
For example, for the polynomial fit question it wasn't clear that our function needed a variable degree parameter.

- The only aspect I can think of is the amount of functions shown during the lectures. It would be useful if more methods were shown in prerecorded videos, as sometimes I had to look for quite a while for a method of accomplishing something in the weekly assessments. Also, I think that the first mistake in each assessment question should be penalty-free. At times it was hard to spot a particular error in the code and so the overall mark was decreasing (I did use the test console and python IDE on my desktop, but sometimes, even if everything was alright in these consoles, MoodleX returned errors).

- Perhaps longer notes, so you don't have to sift through the video to find a particular line of code that you forgot

- The instructions in the assessments were sometimes vague, meaning they were open to misinterpretation leading to incorrect code. Also, more description into how the code is tested would be valuable.

- While the error warning the XMoodle produces can often tell me what particular aspect of the code is wrong; I was often finding myself having to search on google to remind me what the error messages actually meant. I think the course would benefit from teaching students how to handle errors. Introduce python's try and except functionality(Which in my opinion offers an alternative to the insistence function), an accessible table of common python errors with some explanation and an algorithm(Human one) on how fix errors better. I spent a large chunk of time trying to find what my error was and how to fix it. For Numpy arrays, I could not find out how to distinguish between Numpy arrays and matrices without deliberately causing an error and catching it before XMoodle caught it (using try and except). Perhaps make it clearer how the is-instance and type functions can be used to differentiate between the two. Another thing is that it was sometimes difficult to find the text files the worksheet exercises point to.

- with assessments give some examples for us to test with our own code. This was done on some exercises, but not with others. Eg last Q on assessment 5, give us a matrix to test

- Moderation when looking through code. You should get marks for good code as one small thing can lose you all of your marks.

---

**Any comments on what would make workshops more useful and attractive for you?**

- I was managing all the assessed exercises without going to the workshops. I had the impression the workshops were more to support those who were struggling.

- Didn't attend any of them as never had any major struggles with the content. It was reassuring to know we didn't have to attend rather than forcing students to attend something we would get nothing out of.

- Having in person teaching (although obviously not possible in the current climate).

- Didnt attend

- no

- I did not attend the workshops for two reasons,
1.Time Constraints:
The live lectures for PX120 and PX149 all occur on Wednesday morning. I therefore spend Monday and Tuesday doing the module work for those courses as the live sessions assume you have gone through the work for that week before hand. This meant I had no time to attend the workshop sessions.
2. Usefulness:
I was able to solve all the problems in the XMoodle assessment after sorting out the errors I often ended up with. The recorded videos, code notes and slide sheets (they were good and concise) provided enough information to answer most of the questions. I therefore didn't think it workshop was a necessity.
Solutions:
In my honest opinion, the workshops are like the online office hours, not many people show up to them. This is mainly because something else is on, or the workload that day is just to much. Sadly a lot of people will do the bare minimum to pass so keeping it as an option is always going to mean a low turnout. When I had a problem the notes didn't provide a clear cut answer to I just searched up documentation as the recorded videos instructed and if that didn't work I emailed an instructor. This is less time consuming then waiting till next Monday/Tuesday to ask someone via video chat.
I think it comes down to asking people when workshop times would be better placed on the timetable. I also think there needs to be some course related merit, for attending otherwise time constraints will mean people will prioritise what needs to be done.
I know that my solutions are vague but is hard to think of something when in my opinion the Recorded Videos, Code Notes, Slide sheets and the internet the make the workshops a less viable option when your on the clock.

- A little earlier in the learning week, maybe Fridays

- -

- None

- N/A

- going through a challenging set of programming tasks only during the workshop, that explore aspects of programming that are more difficult than what is presented in the lectures.

- I personally never felt like i needed help so i simply never attended the workshops. If I was struggling i am not sure if i would've attended the workshop as I am allergic to all human contact.

- I think the reason I didn't use them was because I didn't have trouble with questions that took me more than a day to figure out.
Maybe a 'compulsory' live lecture with the people involved in the workshops at the beginning of the module would have made people more comfortable to join them?

- Did not use the workshops

---

**Any other comments:**

- no

- Can be difficult to test the assessed problems - to me, coding is all about trying something and testing it, and then fixing what doesn't work. I'm not sure exactly how, but when example data, or something similar, was given, it was very useful.
Thank you

- I really enjoyed this module; coming from having no experience has been a challenge but I feel a lot more confident with programming now

I really enjoyed this module, coming from having no experience has been a challenge but I feel a lot more confident with programming now.

- I really liked this module. I think that it was a perfect 6-CATS addition to other physics and mathematics modules. The amount of work was on point, as it fitted perfectly into my weekly learning scheme. I think I will certainly find some of the methods shown during this module useful in the future. Additionally, it was a great python introductory course. Now, I feel ready to extend my programming skills further. Best regards

- I needed help twice and I got it straight away in no more than 5 minutes thanks to the workshops. Very useful !

- The link for downloading Spyder didn't work for me personally, app on Microsoft store worked though.

- I really enjoyed this module, looking forward to computational physics next year.

- Keep this module how it is in my opinion.

I enjoyed the module.

This is course is a well taught introduction to python. It does not weigh itself down with syntax, it focuses on the problem solving aspects first and