

PROGRAMMING WITH PYTHON

A FEEDBACK TO THE STUDENTS' FEEDBACK

Vedran Šego <vsego@vsego.org>

27.2.2015.



"This was in the suggestion box.
What do you think it means?"

MANCHESTER
1824

The University of Manchester

M\crNA

Manchester Numerical Analysis

OUTLINE

- 1 OVERALL
- 2 THE GENERAL FEEDBACK
- 3 TESTS
- 4 THE LECTURER'S FEEDBACK ☺

OVERALL IMPRESSION

- 22 students turned their feedback in,
- very positive feedback on the materials,
- positive feedback on the help in the lab,
- most think that the course is hard, but interesting,
- a few think it's too difficult,
- a few think it's just right.

1 OVERALL

2 THE GENERAL FEEDBACK

- The time schedule
- The amount of work
- The materials

3 TESTS

4 THE LECTURER'S FEEDBACK ☺

“NOT ENOUGH TIME BETWEEN LECTURES AND LAB CLASSES”

I agree, but this is beyond my control.

Try to go through the exercise problems prior to lab classes as much as the time allows it. At the next lab class, ask about the problems you had difficulties with. Then go through the lab class problems and, again, ask for help. Complete the unfinished problems at home.

“THE MATERIALS ARE PUT ON THE WEB ON THURSDAY”

No, they are put on the web on Monday, three days before the lecture.

“PROVIDE SOLUTIONS ON FRIDAY”

From now on, they will be published earlier, most likely on Saturday. However, **do try** to solve the problems before reading the solutions, as that is the only way to learn how to solve the problems on your own.

“TOO MANY PROBLEMS IN EXERCISES/LABS; WE HAVE TO FINISH THEM AT HOME”

A lot of work at home is a general idea, as emphasised in the introductory lecture. One cannot learn programming in mere 36 hours of lectures and lab classes.

Working on your own is irreplaceable!

“TAKES TOO MUCH TIME FOR ONLY 10 CREDITS”

As per [The University web site](#):

As a general rule, one UK credit equates to 10 hours of work; a 10-credit course unit therefore requires 100 hours of study on average.

Students should note that the number of credits associated with a course does not relate to the number of contact hours (lectures, tutorials etc) a week as is the case in some education systems.

100 hours in 12 weeks is roughly 8 hours per week, in which the classes and the exams are not included.

Please, keep in mind: there are no exams on this course after the lectures end, so it is natural that the work is somewhat compressed in comparison to the courses with more traditional examinations.

“PRODUCE PDFs INSTEAD OF HTML“

IPython Notebook (IPYNB) was chosen for its interactivity which PDFs do not offer. Unfortunately, there are no good converters from IPYNB to PDF files. However, for most students, the HTML seems to work fine.

You can try opening the HTML in a different browser, or using MS Word or a similar tool to adapt the HTML to your needs.

“THE SOLUTIONS ARE SOMETIMES TOO GOOD/ADVANCED”

As the docstrings in these solutions state, most of them are so called “Pythonic solutions”, which aim at those who want to learn more of Python.

These solutions are easily recognized by their file names in the form “name-xx-yy-py.py” or, in rare occasions, some suffix other than “-py”.

They are in no way considered “standard”, and the extra features used there will not be a part of the tests until it is covered in the materials.

All the tests can be solved using only those Python constructs that were shown in the lecture notes and standard solutions (the programs named “name-xx-yy.py” with no suffixes) from the weeks before the tests themselves.

“THE PACE IS TOO FAST”

The amount of the materials is fixed, as is the 12 weeks period.

The materials could be presented a in a bit slower manner (albeit, not much slower), but then you would get far less time for your coursework.

The current intention is to make it possible for you to do the coursework during the Easter break or after it (the submission deadline is Friday in the twelfth week).

“LECTURES ARE USELESS BECAUSE GOING THROUGH SOMEONE ELSE’S CODE IS POINTLESS”

As the only means of learning, yes – it is pointless to just go through someone else’s code.

The purpose of the lectures, however, is to show the syntax and the concepts, and to emphasise the important details. The real learning is done by practicing, i.e., at home and in the cluster.

“LECTURES SHOULD BE GIVEN IN A CLUSTER, SO WE CAN TRY THE STUFF AS YOU SHOW IT”

In an ideal world, this would work nicely. However, there is a time limit of one hour per each lecture, in which we would have to cover the same materials as we are doing now.

Of course, it is paramount that you try the code that is shown, but you have to do so at home, ideally on the same days that the lectures are given.

OUTLINE

- 1 OVERALL
- 2 THE GENERAL FEEDBACK
- 3 TESTS
- 4 THE LECTURER'S FEEDBACK ☺

“TESTS ARE TOO HARD”

This was the first test:

- **Problem 1** is an easier half of Problem 8 from the previous week's lab classes.
- **Problem 2** is Problem 9 from the previous week's lab classes, with the addition of one multiplication with a real number, one `if` statement, and a minor twist in printing: it requires all the numbers to be printed instead of just the last one, which effectively means moving the `print` inside the aforementioned `if` statement.

Like it was said on multiple occasions, including right before the test, you are allowed to use all the course materials, including the solutions of the said problems (which you are allowed to copy and modify as needed).

“WE WANT 2 OUT OF 3 POINTS ON ALMOST CORRECT SOLUTIONS”

The marking is done as follows:

- 1 point for docstrings (all the programs and functions have to have one, explaining – shortly – what the program/function does and what are its parameters),
- 2 points for the correctly working solution; 1 point is given if the solution is close to the working one.

So, yes, you already get 2 out of 3 points for an almost good solution with the required docstrings. However, the programs that don't work at all (for example, those with syntax errors) are not “close”, since they don't do anything.

“THERE IS TOO LITTLE TIME”

OK. From now on, you will have a full hour to solve your tests.

However, keep in mind that no time extensions can replace being prepared and familiar with the course materials.

OUTLINE

- 1 OVERALL
- 2 THE GENERAL FEEDBACK
- 3 TESTS
- 4 THE LECTURER'S FEEDBACK ☺

THE LECTURER'S FEEDBACK (1)

Lab classes are the best place to ask questions – use that!

Some students read the lecture notes for the first time during the lab classes, which is not a good use of the time spent there.

Reading the notes at home, as well as trying to solve at least a few exercises prior to the lab classes, will get you much more prepared to ask the questions about the things that you find hard, thus helping you save huge amounts of time that you'd otherwise have to spend figuring out those same things on your own.

THE LECTURER'S FEEDBACK (2)

Some students spend lots of time in the lab classes talking among themselves or doing things unrelated to Python.

While it is your free choice to do so, you are missing on a precious opportunity to ask experienced programmers and educators for help, which can significantly cut your learning time and save you hours.

THE LECTURER'S FEEDBACK (3)

Some students get stuck, but don't ask for help until one of us notices. While we do try to keep a watchful eye and we are constantly walking around the cluster looking for the students who need help, there is almost 50 of you and only 5 of us. **Please, do ask for help!**

We are here to help you and we are happy to do so.

Remember: asking questions during the lab classes is the easiest way to resolve any problems you have with the course materials!

Of course, you are also welcome to visit during the office hours. ☺

Thank you all for your feedback!

It is truly appreciated.