

MATH20622 Programming with Python - Lab test procedure

Posting of the problems

The test problems will be posted on Friday shortly before 9am on the course website

<http://personalpages.manchester.ac.uk/staff/stefan.guettel/py/>

Once the problems become available, you can start working on them. Note that this is an examination and hence the examination policies apply

<http://www.tlso.manchester.ac.uk/map/teachinglearningassessment/assessment/sectiond-theprocessofassessment/policyonexaminations/>

You can use all materials on the course website, as well as internet resources. You are not allowed to discuss with fellow students or ask for help from the assistants during the test.

Structure of the test

The first lab test will take place on the Friday of week 4 from 9-10am. The relevant material will be the one from week 1 and 2, including the exercises and lab class exercises posted on the course website.

There will be two problems, each of which should be solved in a separate .py file. Make sure

- To start by creating a separate .py file for each problem (for example, create files problem1.py and problem2.py in your week3 folder). Save your files repeatedly as you work on the problems to avoid data loss! If your code is lost you can't get any marks for it.
- To read the problem description carefully before you start coding. The description will tell you precisely what your program should do.
- To start with the important core of each problem, i.e., the main code that solves each problem. You will be given a mark on the code structure and may lose it if your code does not have the essential ingredients required for solving the problem. Hence, for example, do not initially waste time on formatting your inputs or outputs. However, do make sure to spend some time on the proper documentation of your code, as this will be worth a mark (see below).
- To re-read the problem fully when you think you're done. It's upsetting to have a well-written code but it fails to behave exactly as wanted just because a small detail was missed. Note that only programs that behave as specified can get the full marks (there is a mark for code perfection; see below). This is like in real life: You don't want your mobile phone software to crash just because the programmer has not accounted for all eventualities!
- To not chat with fellow students about how you solved the problems until everybody's work has been graded.

Marking

The marking will take place in the second half of the lab class between 10-11am. The markers only have a limited amount of time for each student, so make sure that you are ready for marking after 10am. The marking will be done in randomized order, so when your name is called out the marking starts, and no later. If you're done with solving the problems earlier then you can let one of the assistants know and we can grade your work earlier.

Each problem is worth 0-3 marks. The marking scheme is as follows:

1 mark for proper documentation, i.e., a docstring in every file and function, and if applicable some more comments in the code. Also you need to be able to describe briefly what you've done, and your documentation can help you with this.

1 mark for the code structure, showing that you have the essential ingredients in the code to solve the problem and that the code can be executed without error messages or crashes.

1 mark for code perfection, i.e., your code behaves exactly as required. The assistants will have a number of test inputs to check your code. Only if all these tests are passed you can get this mark for code perfection.

Note that you are not allowed to make any further changes to your code while you are assessed, nor will the marker have time to help you debugging your code.

After the test

You are invited to work on the next problems posted on the course website. Please do not chat with fellow students about how you solved the test problems until everybody's work has been graded.