Summary results of the Universitat Oberta de Catalunya (UOC)

Course

First of all, it is worth stressing that the UOC is a fully online university, therefore all its courses are online. In order to perform the pilots for the QPED project, UOC chose one course, specifically, M0.152 – Programming for bioinformatics (PB), which has 6 ECTS, belongs to the Master's degree in Bioinformatics and Biostatistics and has, on average, 200 students per semester. PB has two runs: the first one starts in the second half of September and finishes at the end of January, whereas the second one starts in the second half of March and finishes at the end of June. Likewise, PB has around 4-5 virtual classrooms which each one has one different teacher who guides the learners and grades the assignments.

PB is the first programming course for most of the students who are enrolled in. More specifically, its table of contents includes:

- Introduction to Python
    - Basic syntax
    - First program
    -
- Key concepts of Python
    - Flow statements: conditionals and loops
    - Functions
- Scientific libraries for Python
    - Math libraries
    - Visualization libraries
- DNA, RNA and sequences with Biopython
- Testing and software quality
    - Unit tests
    - Tests for bioinformatics problems
- Brief introduction to object-oriented programming.

Regarding the assessment policy, PB has 5 graded assignments (GA) that students submit throughout the semester. Each assignment is a Jupyter Notebook with different exercises that students must solve (Some of them are theoretical questions, but most of them are coding problems). The final mark of the subject is calculated by using the following formula: 10% GA1 + 10% GA2 + 30% GA3 + 25% GA4 + 20% GA5. The active participation on the forums is 5%.

Baseline stage

Regarding the baseline, the course started on 16/02/22 and finished on 24/06/22. In order to collect data, the common 4-point rubric created by the QPED group was used for assessing assignments and the diagnostic test. In this regard, only the assignments of one classroom (24 students) were graded by using the rubric (students were not provided with the rubric scores). However, all the students (i.e. 144), regardless of the classroom, performed the diagnostic test. The fifth assignment (i.e. GA5) was used as diagnostic test, since it covered concepts related to testing. Finally, due to technical problems, the questionnaire about students' awareness of software quality was not sent.

Coursework

Firstly, it is worth stressing that GA1 is a simple activity that is used so that students get familiar with the subject, the programming environment and Python. Therefore, this activity may often have higher scores than the others.

"Output correctness" (i.e. the program behaves as expected when the input is correct) and "program robustness" (i.e. the program does not crash/fail when the input is incorrect) are the two items evaluated in most of the assignments and the ones, as it will be shown, which obtained greater improvements. Other items –such as "modularity" and "data type selection"– also improved, but to a lesser extent. For this reason, this short report focuses on such two items (if you need more information, you can ask for the extended version of this report).

As for "output correctness", the figures show a stable trend throughout the assignments:

GA1		GA2			GA3			GA4		
3p	**4p**	**2p**	**3p**	**4p**	**2p**	**3p**	**4p**	**2p**	**3p**	**4p**
22.7	**77.3**	4.8	42.9	**52.4**	4.5	**50**	45.5	20	35	**45**

Table 1. "Output correctness" in GA1-4 during the baseline stage. Percentage of students that achieved each score in each assignment.

Regarding "program robustness":

GA2			GA3			GA4		
2p	**3p**	**4p**	**2p**	**3p**	**4p**	**2p**	**3p**	**4p**
9.5	**47.6**	42.9	4.5	**59.1**	36.4	30	**40**	30

Table 2. "Program robustness" in GA1-4 during the baseline stage. Percentage of students that achieved each score in each assignment.

Diagnostic test

The diagnostic test was the last assignment of the course. Some exercises involved creating tests, writing examples of tests cases (i.e. input-output).

Regarding the "Output correctness" item, the results were:

Diagnostic test (GA5)			
1p	**2p**	**3p**	**4p**
2.6	15.8	29.8	**51.8**

Table 3. "Output correctness" in diagnostic test (GA5) during the baseline stage. Percentage of students that achieved each score in each assignment.

As for “Program robustness”:

Diagnostic test (GA5)			
1p	**2p**	**3p**	**4p**
2.6	17.5	34.2	**45.6**

Table 4. “Program robustness” in diagnostic test (GA5) during the baseline stage. Percentage of students that achieved each score in each assignment.

Validation stage

During the validation phase, the course had few changes. In fact, only the four assignments (GA1-GA4) changed, since they included the TILE approach. More specifically, those exercises that asked students for coding provided learners with a few tests created by using pytest. In some other cases, examples of input and expected output were also given. Likewise, students had access to optional/complementary exercises on Quarterfall platform.

222 students participated in the validation stage from 28/09/22 to 29/01/23. In order to collect data, the common 4-point rubric created by the QPED group was used for assessing assignments and the diagnostic test. Once again, the fifth assignment (i.e. GA5) was used as diagnostic test. Likewise, the questionnaire about students' awareness of software quality was sent to all the students.

Coursework

The results for “output correctness” were:

GA1				GA2				GA3			GA4			
1p	**2p**	**3p**	**4p**	**1p**	**2p**	**3p**	**4p**	**1p**	**3p**	**4p**	**1p**	**2p**	**3p**	**4p**
1.0	2.0	17.7	**79.3**	3.6	8.6	22.3	**65.5**	0.5	15.9	**83.6**	1.1	8.5	26.1	**64.4**

Table 5. “Output correctness” in GA1-4 during the validation stage. Percentage of students that achieved each score in each assignment.

As for “program robustness”:

GA2				GA3			GA4			
1p	**2p**	**3p**	**4p**	**1p**	**3p**	**4p**	**1p**	**2p**	**3p**	**4p**
3	9.1	15.2	**72.6**	0.5	16.4	**83.1**	0.5	11.2	25.1	**63.1**

Table 6. “Program robustness” in GA1-4 during the validation stage. Percentage of students that achieved each score in each assignment.

Diagnostic test

As for “output correctness”:

Diagnostic test (GA5)			
1p	**2p**	**3p**	**4p**
2.2	8.1	15.6	**74.2**

Table 7. “Output correctness” in diagnostic test (GA5) during the validation stage. Percentage of students that achieved each score in each assignment.

Regarding “program robustness”:

Diagnostic test (GA5)			
1p	**2p**	**3p**	**4p**
1.6	8.1	16.1	**74.2**

Table 8. “Program robustness” in diagnostic test (GA5) during the validation stage. Percentage of students that achieved each score in each assignment.

Questionnaire

50 students filled in the questionnaire at the end of the course. They were enrolled in UOC for an average of 1.5 semesters. 54% of the participants stated that they did not have prior experience on programming before starting the semester. The rest (46%) had some basic skills; we say “basic”, because if they had been advanced, the course would have been recognized.

A set of questions asked students (using a 5-point Liker scale) how capable they felt of performing a series of programming tasks after passing the course. As shown in Figure 1, most of students felt very or totally capable of performing the different tasks.

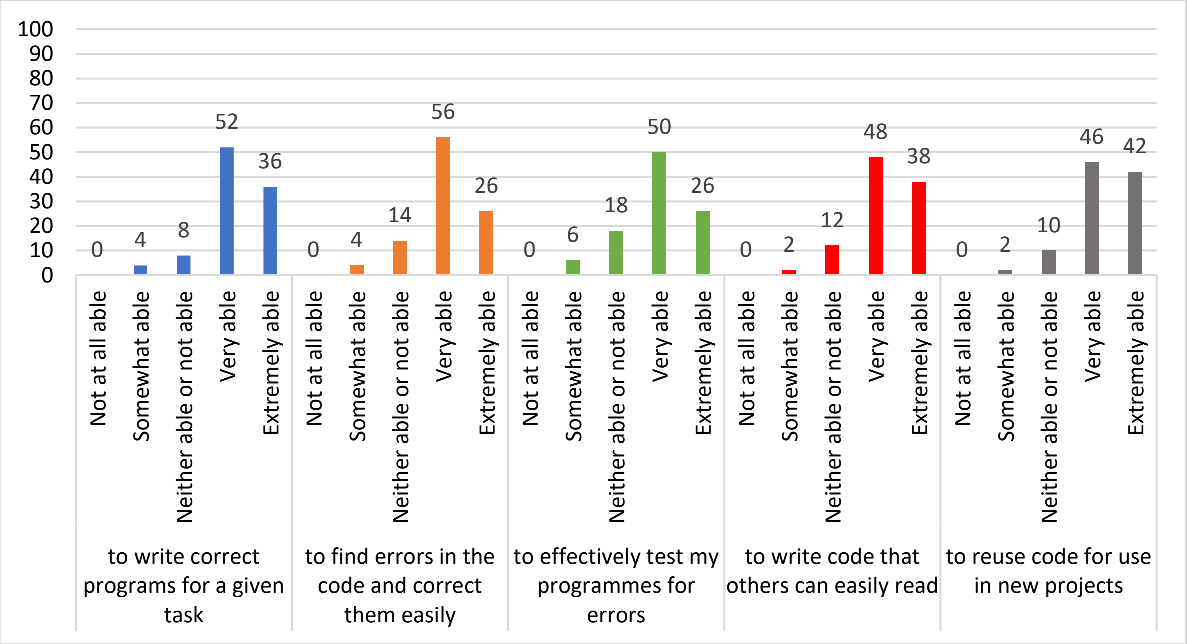

Figure 1. Percentage of answers for “How well do you think you are able…?”

Another block of questions was related to perception, namely to what extend students agreed with different statements on testing. The responses (see Figure 2) reveal that the TILE approach really contributes to a deep awareness of the importance of testing.

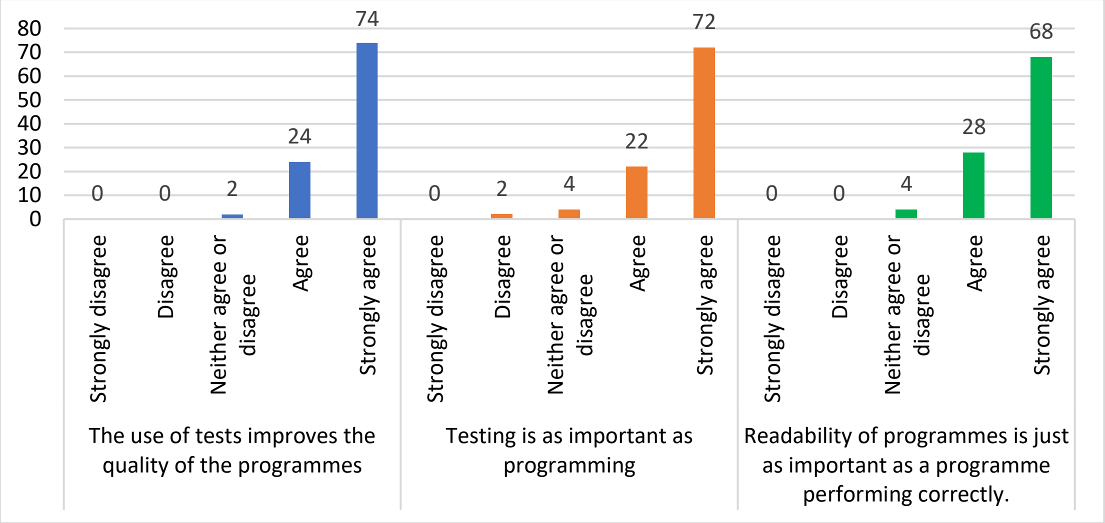

Figure 2. Percentage of answers for “level of agreement with the following statements”.

A closer analysis, which distinguishes responses according to the students’ previous programming experience, shows those students with experience thought testing is as important as programming. This may occur because the experience helped them to notice that programs must be tested properly before handing them in. However, both groups agreed with the fact that tests improve the quality of the programs.

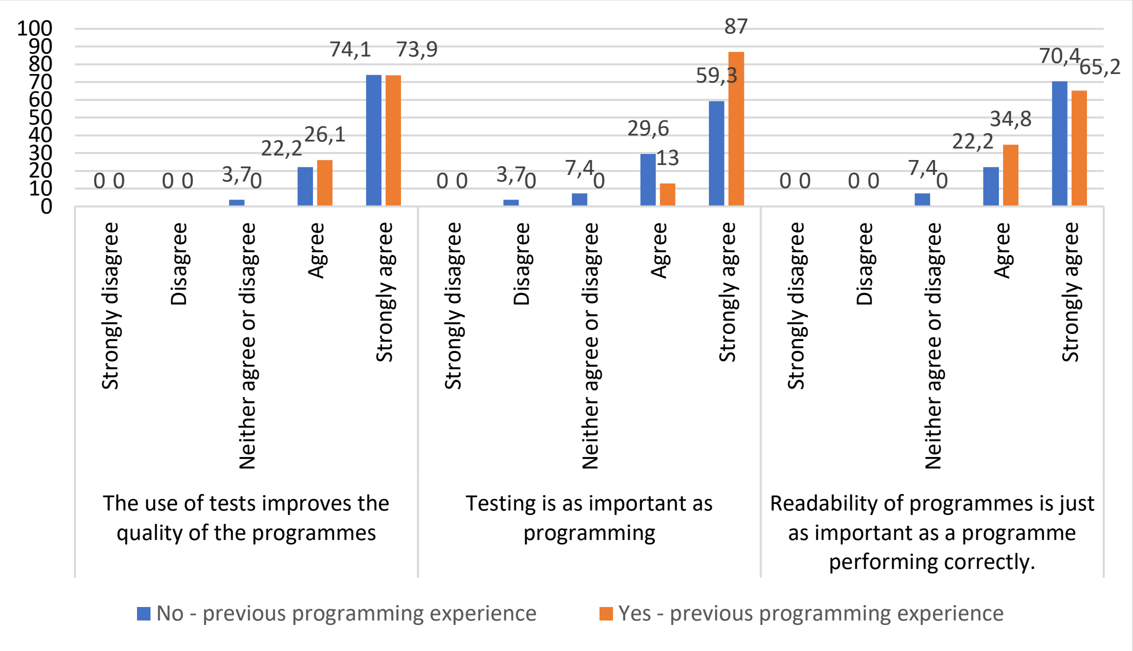

Figure 3. Percentage of agreement with the importance of testing and readability.

The questionnaire also asked about students' programming habits. Figure 4 shows the percentage of students with and without previous experience that scored 4 (rather much) or 5 (very much). As seen, all students are used to running different test cases, it may be due to the fact that the teaching staff, thanks to the TILE approach, has always provided students with different tests cases (i.e. happy and unhappy paths). However, students that have faced more programming tasks are more aware of the importance of testing code thoughtfully. This may mean that students with non-experience only test the most important functions or parts of the program.

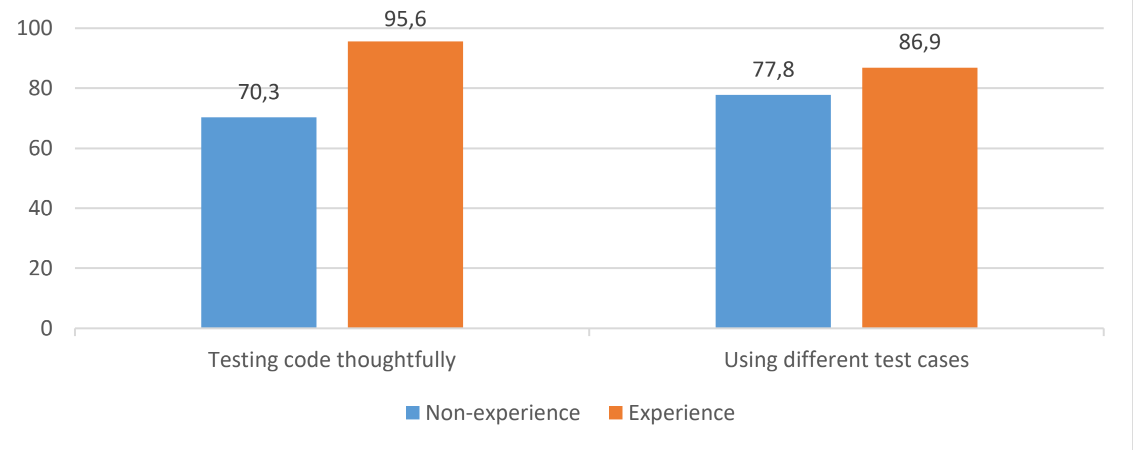

Figure 4. Percentage of students that testing code and use different test cases rather much or very much.

Finally, students were asked about the use of the debugger and tools to check conventions and styles (e.g. flake8). Unfortunately, the results (see Figure 5) show that most of the students did not use such tools. So, these are important issues related to quality software that the QPED project has not been able to address. Therefore, further work must be done.

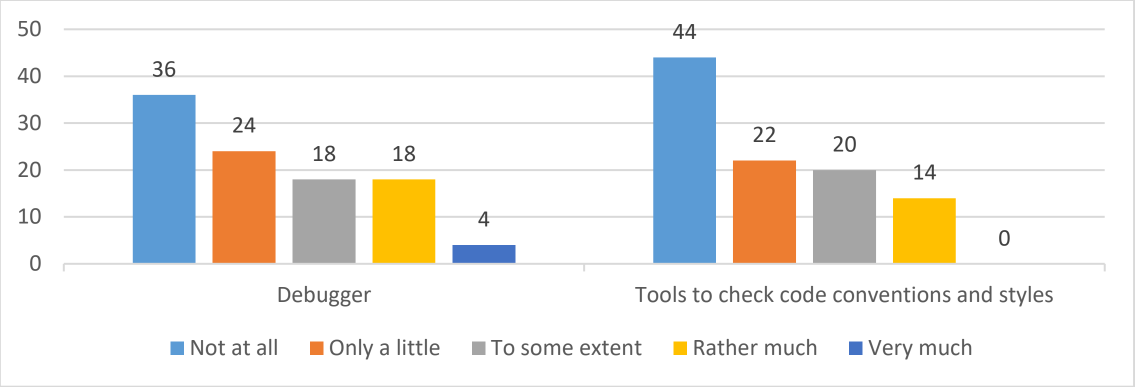

Figure 5. Percentage of students that uses debugger and tools to check code conventions and styles.

Comparison between the baseline and the validation

Regarding "output correctness", we can see the following improvements (in the form of increases) in the score equal to 4:

GA1	GA2	GA3	GA4	GA5 (diagnostic test)
4p	**4p**	**4p**	**4p**	**4p**
+2	+13.1	38.1	+19.4	+22.4

Table 9. Increases in the score equal to 4 for "output correctness" during the validation stage in contrast to the baseline stage.

According to the Chi-square (<0.05), Kolmogorov-Smirnov (<0.05) and the Mann-Whitney U (<0.05) tests, the distributions and measures of central tendency of these this item in assignment GA3 and the diagnostic test (GA5) are statistically different. According to the Gamma positive value, scores in the validation phase were higher than those obtained during the baseline phase.

As shown in the following table, the greater difference between the results obtained in the different stages was on "program robustness".

GA2	GA3	GA4	GA5 (diagnostic test)
4p	**4p**	**4p**	**4p**
+29.7	+46.7	+33.1	+28.6

Table 10. Increases in the score equal to 4 for "program robustness" during the validation stage in contrast to the baseline stage.

According to the Chi-square (<0.05), Kolmogorov-Smirnov (<0.05) and the Mann-Whitney U (<0.05) tests, the distributions and measures of central tendency of these this item in all the assignments (i.e. GA2, GA3 and GA4) and the diagnostic test (GA5) are statistically different. According to the Gamma positive value, scores in the validation phase were higher than those obtained during the baseline phase.

In the light of the results obtained for "output correctness" and "program robustness", we can see that the use of the TILE approach was effective in order to improve students' skills. The fact of providing learners with tests from the beginning may make them be aware of the importance of thinking and evaluating different cases (e.g. look at what happens when the input is incorrect), not only happy paths (i.e. they look at how the program works under perfect and expected conditions). Thanks to the provision of tests, learners get familiar with their format and finally they are able to write good tests on their own. Generally speaking, "output correctness" and "program robustness" are two sides of the same coin. When a program behaves correctly in all the cases (i.e. correct and wrong input), then it is robust.