

# A Teacher's view about introductory programming teaching and learning – Portuguese and Macanese perspectives

Anabela Gomes<sup>1</sup>

Engineering Institute of Coimbra  
Polytechnic of Coimbra  
Portugal  
anabela@isec.pt

António José Mendes, Maria José Marcelino

<sup>1</sup>CISUC – Department of Informatics Engineering  
University of Coimbra  
Coimbra, Portugal  
toze@dei.uc.pt, zemar@dei.uc.pt

Wei Ke, Sio Kei Lm, Andrew Siu

Public Administration School  
Polytechnic Institute of Macao  
Macao, China  
wke@ipm.edu.mo, marcusim@ipm.edu.mo,  
kmsiu@ipm.edu.mo

**Abstract—** The difficulties faced by students and teachers in learning and teaching introductory programming has been a research issue over the years. Demotivation is common in many novice programming students, who are not able to cope with the natural difficulties associated to programming learning. It is up to the teacher to find strategies to help students and keep them motivated during the course. The objective of our research was to know more about the pedagogical and motivational strategies used by teachers in the author's institutions to promote programming student's motivation and learning. Some time ago we interviewed a few Portuguese teachers with diversified experiences in programming teaching. Recently we had the opportunity to do the same kind of research with Professors from Macao (China). The problems identified, the teachers' motivation to teach programming, the educational and motivational strategies and the student-teacher relationship were specifically addressed. This paper describes the research done, stressing the Macanese teacher's views and relating them with the views previously expressed by their Portuguese colleagues.

**Keywords—** *Programming education; Motivational strategies; Pedagogical strategies*

## I. INTRODUCTION (HEADING 1)

The literature includes many references to the difficulties most novice students feel in introductory programming courses throughout the world [1-10]. Diverse factors have been pointed out as causes of student's difficulties. Several authors associate this difficulty with the abstract nature of programming [11, 12-15]; some other authors state that the problem-solving nature of programming also causes difficulties [12]. It is also possible to find other studies that suggest the existence of a correlation between the student's programming abilities and their mathematical skills [11, 13, 14, 15, 16, 17, 18]. Several

pedagogical approaches and tools were proposed to help students learn to program [19-27]. However, high dropout and failure rates in introductory programming courses continue to be reported in many higher education institutions [4, 8, 14, 16]. In [28] the authors discussed a set of difficulties associated to programming teaching and learning. They divide the causes for many students' difficulties in three categories: teaching strategies, student's attitudes, study methods and the natural programming difficulties. The study methods often followed by students are also pointed out as a source of problems. Many of them are used to memorization strategies (reading, seeing solved exercises), which are not enough to learn programming. It is also necessary to engage in an intensive problem solving practice, facing the difficulties of each problem and looking for solutions for them. This should be based in previously acquired generic problem solving skills that often are not present. Programming itself is also a cause of difficulties. Its abstract nature and the languages' features and syntax (created for professionals and not for learners) do not facilitate the students' task. For all these reasons, it is hard for many students to have the necessary motivation, especially in a very difficult time of their lives, in many cases coinciding with a period of transition and instability. Only some students have intrinsic motivation that leads them to be strong enough to carry out the programming learning task successfully. The lack of students' intrinsic motivation forces the teachers to find strategies to attract the students and make them eager to learn how to program.

We wanted to know more about the strategies used to teach programming. So, some time ago we interviewed several Portuguese teachers with diversified experiences in teaching the first programming course. The opportunity of doing the same kind of research with Professors of the Polytechnic

Institute of Macau arouse more recently. Therefore, the following section describes the approaches taken by those teachers. The problems identified, the teachers' motivation to teach programming, the educational and motivational strategies and the student-teacher relationship are specifically addressed.

## II. THE STUDY

### A. Involved teachers

Both in Portugal and in Macau, we interviewed teachers who had previous experience in programming teaching.

In Macau, we involved 5 teachers that work in the Public Administration School of the Polytechnic Institute of Macau. Three of those teachers had over 10 years of teaching experience. One teacher had less teaching experience (about 5 years) and another teacher had 8 years of teaching experience.

In Portugal, we interviewed 18 teachers belonging to the Department of Informatics Engineering of the University of Coimbra and the Department of Informatics Engineering of the Superior Institute of Engineering of the Polytechnic Institute of Coimbra. Most of these teachers (17) had over 15 years of programming teaching experience at different levels. Only one of them had less experience (about 2 years). If we consider only the introductory programming course, only 8 teachers had more than 10 consecutive years of experience. The remaining teachers had taught introductory programming occasionally.

All the interviewed Portuguese teachers have a Master Degree in Informatics Engineering and 50% have a doctoral degree. Four of the interviewed Macanese teachers have a Master Degree and one a PhD, all in Informatics Engineering.

All Macanese teachers have been responsible by lectures and labs. The Portuguese teachers' situation is slightly different, as three of them have been teaching only lectures, four of them have been teaching lectures and labs and the rest (eleven) have been teaching only labs.

In the next sections, we will present the main results concerning aspects like teacher motivation for teaching programming, educational and motivational strategies used, assessment methods and the student-teacher relationship in both the Macanese and the Portuguese contexts. As the Portuguese reality was already detailed in (*omitted, blind review*), the Macanese situation will be more detailed and compared with the Portuguese results in a summarized way.

### B. Teacher's attitudes towards teaching programming

In general Macanese teachers declared that they like to teach programming. When asked to rate their preference to teach introductory programming courses in a 0 to 5 (0 being the lowest and 5 being the highest) Likert scale, two teachers indicated 4, two others answered 3, and only one gave a negative answer (2). We asked the teachers to justify their answers.

The less positive teacher was the less experienced one. He mentioned essentially that *"... teaching the first-year students requires more patience, because students easily lose concentration..."*.

The teachers that gave a 3 answer, justified that *"he needs to spend a lot of effort to make students understand the new programming concepts"* and *"It is challenging and interesting to introduce fundamentals and principles to beginners. However, missing the discussion of advanced topics is somewhat boring, and it is even frustrating that sometimes easy contents cannot be conveyed efficiently."*

The two more positive teachers said *"...it is a challenge to teach beginner students, since they don't possess programming mindset. I feel a sense of achievement when I see that they are gradually developing this mindset..."* and *"It is a satisfying experience to introduce students to programming, but sometimes it is also exhausting."*

These opinions are similar to the preferences expressed by the Portuguese teachers, as most of them manifested a preference to teach topics related with their own research area, but said they also liked to teach programming.

### C. Perspectives about programming learning difficulties

We asked the Macanese teachers which, in their experience, were the most important factors that create learning difficulties to their students. The answers pointed out diverse relevant aspects.

The lack of reasoning skills is mentioned as an important factor: *"... those who have never learned programming before don't possess programming mindset and logics..."*. Most of the teachers also pointed out limited problem-solving skills as a limitation factor to many of their students. Statements like *"...They can understand the syntax quickly and use the syntactic rules to express some sequential steps for simple tasks, but they find it difficult to express complex tasks with loops and branches... they don't even know how to start writing a program..."* seem to translate this idea. One teacher clearly mentioned *"Tackle problem solving step by step"* or *"Designing a solution for the problem"*. Another teacher reported that students often show difficulties to understand the problem to be solved and to idealize the results to be achieved, *"It is not easy to explain how the program will run without using a diagram. Most of the first-year students have difficulties on understanding the lectures and questions purely in English."* Student's abstraction skills were also mentioned by the teachers: *"... It is also difficult to divide a program into subroutines and make interactions between them..."*, *"...Abstractions on different levels, e.g. functions/procedure, objects, data structures, threads, etc..."*, *"In general, beginners often lack the ability to abstract things and understand solutions in a hierarchical manner."*, *"If something does not have a visible effect, it is difficult..."*, are examples of teachers views about this problem. However, but in lesser scale teachers also reported other problems that difficult learning: *"Syntax, nesting brackets (), [], {}, ..."*, *"Students cannot read compile-time error, and ignore them..."*, *"...Program state (e.g. variables), and how to change state with control statements to implement some tasks..."*, *"...Complicated API design in system library. Students don't read, and can't read documentation correctly..."* or *"It's also very difficult for students to capture the idea of using changing variables to accomplish tasks, such as accumulation and counting."*

In general, both groups of teachers mentioned similar problems, such as: low problem solving, logical reasoning, and abstraction skills, and many limitations in basic skills, like interpretation, analysis, discussion and critical thinking. To a much lesser extent they also mentioned difficulties caused by the programming languages characteristics. The Macanese teachers didn't report some problems pointed out by the Portuguese teachers such as: low mathematical background, lack of study organization and minimal work habits and lack of intrinsic motivation and self-confidence. However, Macanese teachers mentioned that students have difficulties in understanding the desired result (usually not easy without a diagram), highlighting the importance of making programming visual for their students.

Macanese teachers consider that the most problematic introductory programming topics are: Recursion; Pointers; Loops and Functions/Methods. In general, they feel that the more abstract topics are more difficult to students. Roughly Portuguese teachers share the same idea.

#### D. Teacher's attitudes towards teaching programming

##### 1) Teaching strategies used in classes

The strategies used by Macanese and Portuguese teachers in lectures follow, in general, a very similar approach. It consists on the presentation and explanation of the syntactic details of the programming language in use. The presentation of each concept is usually followed by small examples showing certain syntax aspects leading to correct, incorrect or unexpected results.

When asked about the general approach used in labs, a Macanese teacher said *"the general approach is to start with an example problem, which everybody is familiar with, so that students can understand what they are going to solve. Then I provide an obvious solution described in natural language and flowcharts, so that students can understand what they are going to tell a computer to do. The most important thing in introductory programming is how to think about the solution in a programming style, and how to translate the thoughts into programs, rather than to really work out the solution. I summarize the patterns to solve typical problems, such as counting, accumulating, and condition-controlled vs. counter-controlled loops"*. Two other teachers mentioned that their approach consists in starting with an easy problem and to use flowcharts to show and explain the steps and the logic of the solution. When the students need to translate the flowcharts to programs, it is the time to discuss language notions and notations, and to introduce the syntactic details necessary to implement it with a real programming language. One of the teachers said that he starts immediately with the syntactic details of the programming language. Other teacher said he uses the programming language syntax from the beginning, but trying to limit it to the basic aspects. The idea is to get the students to write something that compiles and runs earlier in their study.

Although we interviewed a larger number of Portuguese professors, creating the possibility of a higher diversity in the pedagogical strategies they used, the fact is that the strategies reported are not very different. In lab classes, most teachers

usually start by solving an exercise and explaining it to students. Some use only verbal explanations, while others also use other materials, like diagrams and summaries. Sometimes, only when they feel necessary, some teachers make a theoretical introduction of the contents necessary to solve the exercise at hand (this activity seems more usual in Macanese teachers practice). All teachers also mention a close monitoring of students' activities. This means that teachers usually adapt the class flow to the student's performance, for example repeating explanations, proposing alternative exercises, highlighting good (or better) performance by weaker students.

##### 2) Useful teaching strategies

We also asked teachers to indicate other aspects that they think might be useful in introductory programming courses.

Macanese teachers highlighted the importance to make short example programs to demonstrate concepts. Some of them considered more useful to use small exercises that are possible to complete during a class, instead of bigger ones (for which there is insufficient class time). They also considered that small group exercises in classes, promote the discussions between the group members, encouraging students to speak out their concerns in front of only a few colleagues, which seems to be easier to those afraid of talking in front of a big class.

The importance of immediate feedback was also stressed by the teachers. They said it is important that students can review what has happened in their programs and analyze the consequences of their actions. Other aspects mentioned were giving examples to show how to build complete solutions from their components, to give solved exercises, to make analogies, and to make summaries of the important topics. Considering the characteristics of their students, they prefer to use exercises that have some type of visual output, such as simple number crunching, ASCII arts or simple drawings. They consider that the use of graphics makes classes more interesting and attractive to students, and helps them to understand abstract concepts. To use problems/examples students are familiar with also helps to reduce question comprehension load. One teacher said that *"...like human languages, students need to read before write..."* and *"...the programming ideas are distinct from students' previous experience in other subjects, so analogies help them to grasp the abstract ideas..."*. They also highlighted the importance of using visual supports like diagrams or flowcharts to help their explanations. One teacher also reported that after trying many methods, now he reports better results with the creation of twelve logical modules to solve common introductory problems (counting, selecting, ...). He asks students to study those modules and encourages them to map new problems into these logical modules, adopt the appropriate one, and make their development from that basis. Some teachers also consider important to present common errors in programming, and teach students to decrypt compile-time error messages.

The Portuguese teachers' opinions were not very different from the Macanese colleagues. However, they also stressed the importance of creating solutions gradually, with additions and corrections, showing that even the teacher makes errors before reaching a correct solution. They also mentioned that it is important to consider student's pace and to involve students as

much as possible in classes, for example asking one of them to explain his/her solution to the colleagues.

It was curious that even though both groups of teachers point out several useful methods, they are not very different. The main difference was the importance that Macanese teachers gave to visual elements, namely flowcharts, contrasting with a low reference by Portuguese teachers.

### 3) *Student's preferences (teacher's view)*

We also asked the teachers about the strategies/approaches they think are more appreciated by the students. The answers were very similar in both groups: to give examples, to give solved exercises, to make analogies, to summarize. In the teachers' views, students appreciate that presentations are followed by practical activities, so that they can get a full picture of more abstract topics and reinforce their understanding. Most teachers of both groups also mentioned the importance of a friendly relationship with students and a good classroom environment.

### 4) *Teaching strategies to stimulate student's preparation to classes*

Teachers of both groups were asked how they help students to prepare for next class. Both groups answers focused on: Ask students to read specific parts of the textbooks; to give them a few related problems to think in advance; ask them to solve an exercise that involves the next class topics to create the need for them. Other teachers mentioned that they do not put much effort to help students to prepare for the next class, but prefer that they exercise what they have learned in the last class to consolidate it.

### 5) *Teaching strategies to help students in difficulties*

We also asked the teachers which strategies they use when they notice that a student or group of students is lagging behind their colleagues. Often those students can't even start developing a meaningful solution for the proposed problems.

One Macanese teacher said that he tries to explain the problem as concretely as possible through the use of analogies. He also breaks down the problem into smaller pieces, trying that the students develop solutions to the smaller (and simpler) parts. He also mentioned the importance of teaching debugging skills to the students, so that they can find out what is wrong with their programs. Another teacher added the importance of making connections with other examples previously developed. Other teacher said that the exercises he proposed usually fall into one of the three categories: 1) very close to an example already presented in class, introducing only minor changes; 2) a new problem with plenty of oral discussions about its solution, including the detailed steps; 3) a new problem with a brief explanation, and let the students look for the solution. This teacher also draws schemes and summarizes patterns of code for common exercises. He also adds that usually an exercise is accompanied with a sample input and the corresponding expected output, so that students can test their programs. This teacher also stressed the importance of teaching students to debug programs by setting breakpoints and inspecting the values of the variables. More importantly, he added, "*I tell students to prepare a plan of where to stop and what to look at*". This teacher also said that when students are

in difficulties he tries to know if the student understood the problem and helps him/her to work out the first steps to solve it. Another teacher also said he usually breaks down the problem in a sequence of small steps. Each step gives some marks, and later steps are built upon previous ones. The first few steps are often simple and let the students gain some confidence and motivation to attempt it.

The Portuguese teachers' strategies are not very different. Most teachers in these situations use examples or situations familiar to students, trying to reduce the abstract nature of the problem. Another strategy consists in the use of interrogative methods. In doing so they get the student to clarify the core of his/her question/doubt and bring students to discussion. A clear difference was the strong use of the debugger. Portuguese teachers tend to think that its use in the introductory course would add more complexity (as those tools normally are not user friendly) with reduced results.

### 6) *Teaching strategies for different audiences*

Finally, in this section, we asked teachers if they consider important to adapt teaching in function of the characteristics of the students, namely their background, gender, main study area they follow, and so on.

Most Macanese teachers considered important to have in mind the differences. They consider that students with different skill levels (particularly in maths) and intellectual abilities should be approached differently. For instance, one teacher said "*... If most students don't have basic programming skills, I will start by teaching problem solving skills. If not, I'll just start with simple programs, explained with flowcharts and gradually I go to the syntactic details...*". Some teachers also highlighted the importance of using examples that are familiar to the audience. However, there are also teachers that think that programming has its own way of reasoning and its own mental model that must be developed. So, the teaching methods should not differ much for different audiences, as all of them should develop the same skills.

Most Portuguese teachers are aware of the different student profiles they may have in the same class. They address that situation using a variety of support materials and a diversity of exercises that may be used in any class. However, some teachers also think that students with more difficulties should be aware of that fact, so that they take measures to correct the situation.

## E. *Teachers motivational strategies*

### 1) *General strategies*

One of the most difficult challenges for introductory programming teachers is to keep students motivated, active and participatory in the course work. This is particularly true in the case of average and weak students. So, we asked teachers which strategies they use to keep students motivated during the course.

Macanese teachers said that to give students some bonus questions to add extra-marks for the final grade is an effective strategy. To propose interesting or useful problems is also a good approach, as it raises students' interest to solve them. However, they feel it is very challenging to find adequate

problems, since it is difficult for students to feel interested in introductory programming problems, when there is a strong competition from the apps and online games they are playing nowadays. A teacher also mentioned that grouping together in the labs students with different levels often motivates students, as peer discussion helps them to learn from each other. It is also important to encourage the students saying that the more they participate, the faster and easier they will learn. To motivate students, a teacher stated that he usually shows what a relatively simple program that students are able to do, with some fancy effects, such as complex graphics, high precision computations, handling very large amount of data. *"...For introductory programming, I think students need to know the way to tell computers to do what they are good at – to repeat in a very high speed and to store a lot of information..."*, *"...I try to relate the basics they are learning to what they can achieve in year 3 or 4, or what some well-known applications are doing..."* However, *"...a hard job is to make students feel excited with the 'toy projects' they can solve in year 1..."*. They also mentioned that to motivate students it is important to be friendly with them. One teacher referred that students like teachers to share more experiences with them.

In general, the Portuguese teachers' opinions went in the same direction as those made by their Macanese colleagues. Some also stressed the importance to establish close relationships between students and teachers, and to show that the teacher is there for the students. One of the Portuguese teachers even considered that making the students feel more at ease with the teacher is the best motivating strategy. Other teachers use group competitions and role-playing activities to promote discussion and motivation. To ask a student to propose a solution and have it discussed in class was also said to break the routine of the classroom. A teacher mentioned that he often asks the student who finished the task first to explain the solution to the colleagues, as a way to generate some competition that teacher considers beneficial. A curious difference between both groups of teachers was noted, as many Portuguese teachers mentioned the use of games and competitions in class, while no Macanese teacher mentioned this strategy.

## 2) Motivational strategies for apathetic students

The interviews also included a question about how to motivate week students that do not show any interest in programming.

Macanese teachers gave some examples of how they deal with such students: To ask students questions in class, and let other students help them to come up with the answers; To provide some simple game development or real world coursework to grab their attention; To treat students by their name and follow them more closely during lab sessions and practice work; To show they can accomplish the tasks assigned if they commit to it; To praise for their small progresses and try to let them know that things are not so scaring, once they get into it a bit. A teacher said that *"...Sometimes I talk with them in lab session and get them complete some tasks. Some students need a mentor to hold their hand and walk a few steps in early attempts in programming... programming is intimidating for beginners... hopefully they can stand by themselves later..."*.

The Portuguese teachers' opinions on this issue were not very different. They think that the best way to stimulate the motivation and self-confidence of apathetic students is to approach them, observe what they do, and clear their doubts, until the difficulties are overcome. They consider that it is important to insist with these students, until they get some results, even if they represent only a small progress. To treat the students by their name is also important, as it shows students the teachers have some interest in them.

## 3) Motivational strategies for anxious students

Often teachers should deal with students that are over anxious about the possibility of failure. So, we asked both groups of teachers how they deal with this situation.

Macanese teachers gave the following answers. To give them some extra work to practice; To give them interesting or useful problems to solve; To give them reassurance whenever they are on the right track; To encourage them to do more exercises to make them more confident; To make them aware that there is always a learning curve. One teacher said *"...I would tell them that this is the way it is, difficulties are normal and things tend to improve ... Besides, I would encourage them to try things and don't be afraid of failing... after all, the computer will not blow up..."*. Another teacher said that *"...Actually, I do not have many of these students. Usually those who really care about their scores and outcomes can get through the course very well. It is quite often that they have low marks in the first test, when I generally point out the weakness and they know how to improve following the instructions. I believe, if a student is worrying about the possible failure, following some exercises seriously, she/he will improve and pass..."*. In general, teachers think it is important to try to pass confidence to students *"...I literally tell them 'the computer will not explode if they make mistakes. Just try'..."*.

The Portuguese opinions were very similar considering important to pass confidence for the student and encourage them to keep trying to develop their solutions to the proposed problems.

## 4) Motivational strategies for students with different skills

One of the most commonly mentioned difficulties about introductory programming teaching is coexistence in the same class of students with very different skills. So, we asked how teachers deal with this situation.

Macanese teachers mentioned several possibilities: To give at least one challenging coursework; To encourage students to learn from each other and advise them to do more exercises; To prepare problems with different levels and suggest the weaker students to start by solving smaller problems instead of dealing with more complicated ones; To give good students variations of the original question, as a challenge. Although most teachers consider important that weaker students learn from the best students, they also mention that sometimes the weaker students rely too much on the good students. In other situations the good students are not interested to help the colleagues to learn and they simply let them copy their works, which has a very low educational value. One of the teachers said that, usually, he does not use a different approach for students with different levels. However, when they raise questions orally or by email, he gives suggestions individually, based on their questions and

level. He also suggests different approaches to all students in the classroom and encourages them to try each of them. Students discussions about homework are also stimulated.

Most Portuguese teachers said that they give students different assignments, trying to adapt its difficulty level to the knowledge of the students. Other teachers consider that group work could be an effective strategy to reduce this problem, but it may have other side effects, as also mentioned by the Macanese teachers. The constitution of the work groups didn't have consensus among the teachers. Some prefer that the students choose who they work with. Others think that group members should have a similar knowledge level. Still others think that a stronger and more motivated student and a weaker and unmotivated one should be grouped together.

#### F. Assessment strategies

We also asked the teachers which assessment methods they use in introductory programming courses.

Several Portuguese teachers defend continuous summative assessment as the most beneficial methodology. However, due to the high number of students and the associated work overload, they usually only do a final exam. However, other teachers consider that assessment should be used as a pedagogical tool to stimulate students to work. Consequently, they defend that courses should include at least one project with gradually over time assessments, each of them with a specific goal. Other teachers also consider that assessment is essential for students to feel pressure and practice, as nobody learns programming without investing intensive training time. For some teachers, continuous assessment systems can also serve as a motivation mechanism, since the extrinsic motivation generated by the assessment results is very important for many students.

Most Macanese teachers also agree with continuous summative assessment. They usually have 2 closed book tests during the course, a final closed book exam and open book exercises that jointly with the students' class participation correspond to around 20% of the final grade.

#### G. Final Recommendations: The ideal teacher

We also asked the teachers to indicate the characteristics that, in their opinion, make an ideal introductory programming teacher.

Macanese teachers gave the following answers: *"I think the ideal teacher should be friendly, well prepared, and provide visual teaching materials, so that students can understand the concepts easily."*, *"A good teacher to teach introductory programming must have good experience (at least 3 years) to deal with different levels of students, knowing their different backgrounds, and being patient, dynamic and able to relate abstract thinking in a concrete manner."*, *"The ideal teacher should be patient with many years of teaching experience and knowing a lot of interesting background stories."*

The general Portuguese opinions also fell essentially in experienced and patient teachers.

#### H. Final Recommendations: Factors for success

We asked the teachers to indicate the most important success factors in programming learning.

Macanese teachers said: *"Students should be responsive in the class"*, *"Students should read a lot of good examples"*, *"Students should keep on trying things out, not afraid of failing"*, *"Students should be persistent to get a program finished by themselves"*, *"Students should be dynamic and able to think differently"*, *"Students need to have will to practice and study examples"*, *"To learn programming successfully, one needs the ability of critical thinking, the interest in details and the appreciation of subtlety"*, *"When students learn that computers work in a different way from human, and they have patience and interest in writing something computers understand, they will succeed in learning to program"*.

The Portuguese opinions about this topic were very similar, students should practice a lot, be persistent, and shouldn't be afraid of asking teachers and never give up.

It is interesting to note that both groups of teachers put the main success factors on the student's side. Teachers or institutional issues were not mentioned by any teacher.

#### I. Final Recommendations: Factors for failure

After asking about success factors, we asked also about failure factors.

Macanese teachers said that the main difficulties in programming learning result from the lack of interest, the lack of courage to accept errors and failures and the lack of persistence that many students show. One teacher adds, *"...As I often explain to my students, there is a high threshold to shape up a programming mindset. To get through it, students must practice a lot... until they get through it, and then it is possible to advance very quickly..."*, *"...Teaching programming is like teaching a foreign language. Students must read and write to master a programming language, but it would be best if teachers can give personal guidance to students..."*.

The Portuguese opinions also fall in a variety of student related factors: lack of motivation, incorrect study habits and low basic background, namely in problem solving and mathematics.

Again, teachers focused essentially in student related factors.

#### J. Final Recommendations: Programming language and environment to start programming

The choice of the programming language to use in introductory programming course is an old debate among educators. The same can be said about the development environment students should use in those courses. Even though our interview did not ask that question directly, several teachers mentioned it in their answers. There are those who state that some programming languages are more appropriate to start than others. Usually they state that programming initiation should be done with a simpler language, like Python or a language that facilitates the creation of animations and other visually appealing programs, like Processing. Other teachers affirm that the choice of the language should depend on the degree and dominant student profile.

The institutions where the interviewed teachers work are a good example of the diversity in this question. The Macanese teachers use Java in their introductory courses, while the Portuguese use Python and C for Computing degrees (we have teachers from two different Portuguese institutions), and Processing for the Design and Multimedia degree. Some teachers said that they don't think the language used in their institution is the most adequate, but they don't have the power to change it, as that decision is made at institutional level. In most cases the courses use professional integrated development environments (IDE), such as Visual Studio or Eclipse. Although teachers consider they are more complex than necessary for introductory courses, they also say its utilization is not a major obstacle to most students, as they only use basic features that are not difficult to learn. Also, these teachers normally give students a step-by-step tutorial about the important aspects of the IDE in use.

There were also some teachers that indicated a preference to start the courses using paper and pencil programming, especially in labs. The idea is to relieve students from syntactic details and let them focus on problem-solving and algorithm development. We found some Portuguese teachers using pseudocode in initial classes and Macanese teachers using flowcharts for this purpose (and whenever they feel necessary to clarify some algorithm). However, we also found teachers that defend the use of a programming language right from the beginning. They feel that programming on paper is too artificial for students and it doesn't allow the type of testing that is only possible with a computer.

Most interviewed teachers believe that the programming language used is not the most important issue, because the big problem is the students' inability to solve problems. The Macanese teachers' opinion is not very different. However, one of them has a different opinion: *"...I would like to teach programming using a language whose underlying model is already familiar to the audience, for example, assembly languages, with only simple arithmetic operations and jumps, rather than the while-loop abstraction..."*, *"...Another example is using functional languages with direct mathematical models for those who are good at mathematics..."*

### III. CONCLUSION

Learning to program is not an easy task for many students. It is usual to find high dropout and failure rates in introductory programming courses. Although many reasons can contribute to this situation, we still believe that most students will be able to learn if the teacher provides an adequate support, especially in the motivational aspects. The objective of this research was to know more about the strategies used by teachers in different institutions and cultures to promote programming learning and students' motivation. We interviewed teachers with diversified experience in teaching the introductory programming course, in order to understand the pedagogical approaches used.

Even though we interviewed teachers from very different cultures, that work with students also raised in very different contexts, the difficulties and problems identified, the pedagogical approaches and motivational strategies used were

very similar. We also found some consensus about the factors that promote good results. Aspects like the student-teacher relationship, the motivational strategies used, the assessment methods or the materials presented in class were pointed out. Most interviewees considered that the teacher proximity is also crucial to attract students and encourage them to keep on working until they reach a good programming level. Continuous assessment is also considered valuable, although not all teachers declared to use it.

The main differences we found were the heavier use of visual representations, flowcharts and other diagrams, by Macanese teachers. They feel that it is the best strategy to improve students' comprehension. On the other side, several Portuguese teachers stressed the importance of class competitions and challenges or as a form to promote student engagement and interest. Macanese teachers did not mention this type of strategies.

### ACKNOWLEDGMENT

The authors would like to thank all the interviewed teachers for taking their time to answer our questions and giving valuable suggestions and tips to improve programming teaching and learning.

### REFERENCES

- [1] M. Konecki, Problems in programming education and means of their improvement. In B. Katalinic (Ed.), *DAAAM International Scientific book* (Chapter 37, pp.459-470). Vienna, Austria: DAAAM International, 2014.
- [2] A. Carbone, J. Ceddia, D'Souza Simon and R. Mason, "Student Concerns in Introductory Programming Courses," *Proceedings of the fifteenth Australasian Computing Education conference*, Adelaide, Australia, pp. 41-50, 2013.
- [3] J. B. Benmadsen, M. E. Caspersen and M. Kölling, *Reflections on the Teaching of Programming: Methods and Implementations*. New York, USA: Springer Publishing Company, Inc., 2008.
- [4] E. Lahtinen, K. Ala-Mutka and H-M Järvinen, "A study of difficulties of novice programmers," *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, Monte de Caparica, Portugal*. ACM New York, NY, USA, pp. 14-18, 2005.
- [5] R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, et al., "A multi-national study of reading and tracing in novice programmers," *SIGCSE Bulletin*, vol. 36, no. 4, pp. 119-150, 2004.
- [6] C. S. Bruce and C. A. McMahon, *Contemporary Developments in Teaching and Learning Introductory Programming: Towards a Research Proposal*. Teaching and Learning Report. Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia, 2002.
- [7] E. Dunican, "Making The Analogy: Alternative Delivery Techniques for First Year Programming Courses," *Proceedings of the 14th Annual Workshop of the Psychology of Programming Interest Group*, Brunel University, London, UK, pp. 89-99, 2002.
- [8] T. Jenkins, "On the difficulty of learning to program," *Proceedings of the 3rd Annual LTSN Information and Computer Science Conference*. Loughborough University, UK, pp. 53-58, 2002.
- [9] R. Lister, "On blooming first year programming, and its blooming assessment," *Proceedings of the Australasian Conference on Computing Education (Melbourne, Australia)*. ACM New York, NY, USA, pp. 158-162, 2000.
- [10] L. E. Winslow, "Programming pedagogy-a psychological overview," *SIGCSE Bulletin*, vol. 28, no. 3, pp. 17-22, 1996.

- [11] J. Bennedsen and M. E. Caspersen, "Abstraction ability as an indicator of success for learning object-oriented programming?" SIGCSE Bulletin inroads, vol. 38, no. 2, pp. 39-43, 2006.
- [12] P. Byrne and G. Lyons, "The effect of student attributes on success in programming," SIGCSE Bulletin, vol. 33, no. 3, pp. 49-52, 2001.
- [13] A. Gomes, L. Carmo, E. Bigotte and A. J. Mendes, "Mathematics and programming problem solving," *Proceedings of 3rd E-Learning Conference - Computer Science Education, Coimbra, Portugal*, e-learning'06, CD-ROM, 2006.
- [14] A. Gomes and A. J. Mendes, "A study on student's characteristics and programming learning," *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications, Vienna, Austria*. EDMEDIA'08, AACE, pp. 2895-2904, 2008.
- [15] A. Pacheco, A. Gomes, J. Henriques, A. Almeida and A. J. Mendes, "Mathematics and Programming: Some studies," *Proceedings of the International Conference on Computer Systems and Technologies, Gabrovo, Bulgaria*. ACM New York, NY, USA, pp. 77-82, 2008.
- [16] Mathematics Education, Dublin, UK, pp. 228-239, 2007.
- [17] R. Lister, B. Simon, E. Thompson, J. L. Whalley and C. Prasad, "Not seeing the forest for the trees: novice programmers and the SOLO taxonomy," SIGCSE Bulletin, vol. 38, no 3, pp. 118-122, 2006.
- [18] E. Tomai and C. Reilly, "The impact of math preparedness on introductory programming (CS1) success," *Proceedings of the 45th ACM technical symposium on Computer Science Education (SIGCSE'14)*, Atlanta, Georgia, USA: ACM New York, NY, USA, pp. 711-711, 2014.
- [19] "Scratch" [Online]. Available: <http://scratch.mit.edu/>. [Accessed January 2017].
- [20] A. Santos, A. Gomes and A. J. Mendes, "Integrating New Technologies and Existing Tools to Promote Programming Learning," *Algorithms*, vol. 3, no. 2, pp. 183-196, 2010.
- [21] T. Rajala, M.-J. Laakso, E. Kaila and T. Salakoski, "VILLE: a language-independent program visualization tool," *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88 (Koli Calling '07)*, Vol. 88. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 151-159, 2007.
- [22] M. Kolling, B. Quig, A. Patterson and J. Rosenberg, "The BlueJ system and its pedagogy," *Journal of Computing Science Education, Special Issue of Learning and Teaching Object Technology*, vol. 12, no. 4, pp. 249-268, 2003.
- [23] M. Ben-Ari, N. Myller, E. Sutinen, and J. Tarhio, "Perspectives on Program Animation with Jeliot," *Lecture Notes in Computer Science*, 2269, Springer-Verlag, pp. 31-45, 2002.
- [24] E. Roberts, "An overview of MiniJava," *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education (SIGCSE'01)*, Charlotte, USA, 2001.
- [25] T. Naps, J. Eagan and L. Norton, "JHAVÉ – An Environment to Actively Engage Students in Web-based Algorithm Visualizations," *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, pp. 109-113, 2000.
- [26] D. Arnow and O. Barshay, "WebToTeach: An Interactive Focused Programming Exercise System," *Proceedings of the 29th ASEE/IEEE Frontiers in Education Conference*, San Juan, Puerto Rico, pp. 39-44, 1999.
- [27] R. Pattis, *Karel the Robot: A Gentle Introduction to the Art of Programming*, John Wiley & Sons, 1981.
- [28] A. Gomes and A. J. Mendes, "Learning to program - difficulties and solutions," *Proceedings of the International Conference on Engineering Education- ICEE'07 (CD-ROM)*, Coimbra, Portugal, 2007.