Programmer education in Arts and Humanities Course Degree.

Lorella Gabriele
Francesca Pietramala

Centro Interdipartimentale della Comunicazione
francesca_lorella@yahoo.it

Keywords: POP-I.B. preprogramming knowledge, POP-II.A. individual differences, POP-V.B. interviews.

Abstract

In this paper, we present some results of a questionnaire research made at the Arts and Humanities Faculty nearby University of Calabria. We have investigated if, in learning a programming language, students coming from a humanistic background differ from students having a scientific one. Furthermore, we tried to verify if students experienced some learning difficulties and if a potential reason of these difficulties could be attributed to their school background.

Introduction

Since three years, at DAMS (Disciplines of Music, Arts and Entertainment), Arts and Humanities Faculty nearby University of Calabria, has been activated a Multimedia course degree which comprehends a curriculum of basic computer science, among which there is the Programming Languages course. There is neither "experience" in teaching to the Arts and Humanities students (since University have used teachers coming from Computer Science Departments), nor in ascertaining if a humanistic background could influence teaching a Programming Language. Instead, in other countries, there are many research studies in programmer education and a great deal of improvements in teaching Programming Languages at the Arts and Humanities Faculty. We have found interesting some recent papers for our research that are the following:

Good et al. (1994) describe the implementation of two approaches to supporting novices in their attempts to learn Prolog.

Merrill and Reiser (1994) present results from an experimental study that compared learning outcomes of students solving introductory programming problems in three different learning environments.

Merrill and Reiser (1994) describe three groups of actions students have to perform in learning to programme and the outcomes of these actions for novices.

Draper (1996) suggests that should identify explicitly all the component skills that a programmer should have and to design teaching for each of these skills, instead of merely expecting them to be acquired implicitly through practice of the overall activity of programming.

Blackwell and Green (1999) describe three experiments that manipulated the degree of metaphor in Visual Programming Languages.

How do Arts and Humanities students learn Programming language? In this paper, we present an analysis of programming language learning at the Arts and Humanities Faculty, investigating if students experienced some learning difficulties and if a potential reason of these difficulties could be attributed to their school background.

Subjects

The sample are composed by fourteen students: 11 male and 3 female.

The subjects, aged about 19-35 years old, are attending Arts and Humanities (DAMS), Multimedia course degree. The students are following the course of Programming languages, lecturer Professor G. Rossi.

Materials

We have used two questionnaires.

- The first closed answer questionnaire, with twenty-eight questions, is subdivided in to three sections (A; B; C). Section A: has been useful to get information about every student, such as: name, age, sex, and school background. Section B: presents fourteen questions to get information about how student use computer. Section C: presents some questions in order to get information about student's studying methodology, how long time they study, every day and their future expectations.
- The second questionnaire has been done in collaboration with Professor G. Rossi, lecturer of the course of Programming languages and in this questionnaire there are specific exercises such as: " Indicate the type of all variables who appear in a declaration". "Indicate the output of a piece of a program", etc.

Method

At beginning of the course, we administered the first questionnaire. We said to the students that the questionnaire would be useful for teaching and educational aims. The questionnaire had a closed format and the subjects were allowed to select multiple answers; it has been administered in the lesson time.

During the course, we administered to the students the second test and we got information concerning the progress of each student.

We have valued section B of the first questionnaire, allotting one point for each correct answer in order to classify the basic computer science knowledge. We used the following categories:

VERY GOOD (about 12, 14, correct answers), GOOD (about 12, 10, correct answers), DISCREET (about 10, 8, correct answers), SUFFICIENT (about 8, 6, correct answers), INSUFFICIENT (less than 6 correct answers).

The second questionnaire, with specific exercises, has been valued together with Programming Language lecturer. It has been used a numerical method (A1+A2+...+An/25*30 +1;0) and after we translated these results with the categories above mentioned.

Results

Results concerning the first questionnaire.

- Section A:

  Fourteen students compose the sample: 79% are male and 21% are female.

  7% of the sample attended a " High Professional School " and another 7% attended at " Italian High School for Teacher ", while 43% attended at "High Scientific School ", 43% attended at "High Technical School".

All the students of the sample, actually, is registered at Arts and Humanities Course Degree and have a Computer at home that 71% use to work, 79% use to navigate in Internet, 64% use computer for school exercises and to play.

50% of the sample spend more than three hours, a day, with the computer, 7% less than three hours a day and 43% at least one hour a day.

93% of the sample have already followed a Basic Computer Science course.

50% of the sample have used some programming language very short components such as multimedia, HTML, graphics…

- Section B:

In evaluating the basic computer science knowledge of each student, we got that 21% of the sample have a VERY-GOOD knowledge, 57% of the sample have a GOOD knowledge and 21% of the sample have a DISCREET knowledge.

MARK	**RELATIVE FREQUENCY**
VERY-GOOD	21%
GOOD	57%
DISCREET	21%

Table 1 – Results of the first questionnaire: section B, basic computer science knowledge

- Section C:

All the students use a graphic representation to help themselves solving a particular problem, because:

1. "The graphic representation help to represent mentally the necessary passages to the problem resolution" (54%)
2. "The graphic representation simplify the logical reasoning" (21%);
3. "By graphic representation it is possible to order and structure the key-concepts " (57%).

93% of the sample frequent all days this course of Programming Languages, while 7% of the sample does not attend regularly. 79% in order to follow the course easily, repeat each lesson self-governing and exercising themselves practically every day.

Results concerning the second questionnaire:

MARK	**RELATIVE FREQUENCY**
VERY-GOOD	22%
GOOD	14%
DISCREET	7%
SUFFICIENT	7%
INSUFFICIENT*	50%

Table 2 - *Results of the second questionnaire.*

* There are three students that beginning to attend at the course with a week later.

Results' Analysis

We have analysed the questionnaire data and although the sample is very small and standard error is rather high, we have crossed some variables and we report some correlation.

22% of the students that, at the second questionnaire, have VERY-GOOD. This part of the sample has the following characteristics, which we have extrapolate by the first questionnaire: 7% of this value, have a DISCREET basic computer science knowledge and have attended a Italian High School for Teacher; while 15% have a Good basic computer science knowledge and have attended Scientific School.

14_% of the sample that, at the second questionnaire, have got a GOOD mark, confirm the mark of the first questionnaire and have attended a High Technical School.

7% of the students that, at the second questionnaire, have got a Discreet mark, instead, have a VERY-GOOD basic computer science knowledge and have attended a High Technical School.

Another 7% of the sample that, at the second questionnaire, have got a SUFFICIENT mark, have a DISCREET basic computer science knowledge and have attended a Scientific School.

50% of the students, at the second questionnaire, have got an INSUFFICIENT mark. 21% of this part of the sample have begun to attend to the course a few weeks later the start of the course of Programming Languages. Indeed 14% (of this part of the sample) have a VERY-GOOD basic computer science knowledge and have attended a Scientific School; while 7% (of this part of the sample) have a DISCREET basic computer science knowledge and have attended a Professional School; 28% that have a GOOD basic computer science knowledge, have attended a High Scientific School (7%) and High Technical School (21%).

A glance to students' cognitive styles

It should necessary to make intensive experimentation with a more representative sample in order to deepen if these results are true or not. We have decided to search for some other information in order to outline the cognitive styles students present in learning Programming Languages. In fact, at the end of the course, we have realised a structured interview with the same sample of students. We have used some of the questions which were in the first questionnaire. Briefly, we interviewed students about:

- which difficulties they have met during the Programming Language course;
- studying method utilised for this course;
- differences (studying method or learning difficulties) between a humanistic course and a scientific one;
- difficulties they met while they developing an algorithm;
- to state the coffee algorithm, step by step;
- if they use flowcharts or particular scheme to improve their ability in solving an algorithm;
- if they know another programming language and when and why they have utilised it;
- differences between Visual Basic and other programming language, they had learned before.

What we can point out from this final interview?

By the interview's answers analysis, here there are some problems we have drawn-up. We have extrapolated three groups of subjects which have three different cognitive styles. The first group has had some problems in attending and studying the PL course. The second group, that has had fewer problems. A third group which has had no problems. These groups of students roughly correspond to the category we have chosen to describe our sample. In fact, it seems to us there are three students' cognitive styles. In the following we present some observations coming out from the interview.

1. Difference between studying methods:

a) The first group of students has irregularly attended the course of Programming Languages. They have noticed that a specific studying method was necessary: to do exercise every day and after every lesson. They have underlined that a traditional course of Arts and Humanities is easier than a scientific one. In the former, it is sufficient to get information about a particular topic, in the latter, such as Programming Languages, it is required a constant study and a great deal of practice.

b) The second group of students have understood that in order to programme it is necessary to do exercises at home, because the concepts are simple and comprehensible but it is helpful to have a good practice.

c) The students that were already accustomed to formal reasoning, have declared that the course of Programming Languages requires a considerable size of work, it is important to integrate theory and practice and they made exercises all the course long for three or four hours a day.

2. Differences about algorithm developing:

a) The first group of subjects, detects only the steps immediately arguable about algorithm developing or prefer to give no answer.

b) The second group of subjects have a reasoning capabilities deeper compared to the previous group, but they aren't able to specify integrally the complete set of step necessary to complete the cycle.

c) The third group develops an algorithm in a complete way, listing every steps.

3. Use of supporting to problems solving (flowcharts, schemes or imagination):

a) The first group uses no-structured schemes, but only when the problem is very difficulties.

b) The second group uses imagination and flowcharts, but only when the problem require a lot of steps.

c) The third group uses always flowcharts, because so have learned to solve the problems and so is easier.

4. Difficulties about management graphic interface and realisation of the code that management the interface.

a) The first group meets difficulties about basic processes of Computer science: conditional and sequential instructions, iteration processes and selective structures.

b) Visual Basic is a textual language, which use a graphical GUI builder to make programming decent interfaces easier on the programmer. The second group of students have declared to be stimulated to programme by the graphic interface, but they are hindered by the code, because they are not able to imagine what the code could realise.

c) The third group met no difficulties because they have got good basic skills in programming. Stimulated by the graphic interface Visual Basic presents, they have got a considerable mastery of this programming language during the course.

5. Do students know to identify differences among programming languages?

a) The first group of students that attended irregularly at the Programming Languages course was not able to express an opinion about.

b) The second group of students said programming languages of last generation require a mind trained about programming.

c) The third group of students said that there are not so large differences between languages of last generation: what changes is only the way to express the code.

Conclusion

In our opinion, these data, which we have got from our research, are not sufficient to conclude that there is a cause-effect relation between school background and learning a Programming Language. Our sample is very small so is not possible to extend the results at statistical universe. However, we can say that students with a humanistic background have really met greater difficulties than students coming from a scientific one.

Acknowledgements

Thank you to Dr. Eleonora Bilotta.

References

Blackwell, A.F. & Green, T.R.G. (1999). Does Metaphor Increase Visual Language Usability? *In Proceedings 1999 IEEE Symposium on Visual Languages VL'99, pp. 246-253.*

Draper, Stephen W. (1996). Programming skills, visual layout design, and unjustifiably usel testing: Three reports in the psychology of programming, PPIG96.

Good, J. Brna, P. & Pain, H., (1994) Prolog Unification: Diverse Teaching Strategies for Novices*, 6th Workshop of the Psychology of Programming Interest Group (PPIG-6), Milton Keynes*.

Reasoning-congruent learning environments: Scaffolding learning by doing in new domains Merrill, D. C., & Reiser, B. J. (1994). *Technical Report #55, The Institute for the Learning Sciences, Northwestern University*.

Scaffolding effective problem solving strategies in interactive learning environments Merrill, D. C., & Reiser, B. J. (1994) *The Proceedings of the 1994 Conference of the Cognitive Science Society.*