Eeney, Meeney, Miney, Mo?
Selecting a First Programming Language

Leila Goosen [1], Elsa Mentz [2] and Hercules Nieuwoudt [2]
1 Faculty of Education, University of Pretoria, 0001 South Africa
lgoosen@gk.up.ac.za
2 Faculty of Education Sciences, North-West University, 2520
South Africa {snsem,nsohdn}@puknet.puk.ac.za

Abstract. This study established a list of criteria to select a first programming language. We used a literature study to create criteria for developing thinking and programming skills, student appropriate programming learning environments, new tendencies in programming, practical programming issues, affordability, training and resources, and general-purpose programming. An empirical study verified the validity of selection criteria identified.

1 Introduction

With so many languages available for teaching programming at an introductory level [1], choices remain controversial [2]. Computer technology and languages develop at an alarming rate [3], and programmers cover more languages in their lifetime than ever before. While learning practical programming skills in a particular language, it is even more important for novice programmers to develop a sound theoretical understanding of programming, to prepare for later learning future languages and environments. Rapid technology and language changes also mean that selecting the wrong implementation could mean that both its use and its teaching will be outdated shortly. Even if ignoring limitations regarding support, costs, training etc., one needs to select a programming language for use in a first course using different criteria [1].

2 Results

We used a literature study to develop criteria for selecting a first programming language. Further details regarding the research method and the instrument used can be accessed via [4]. Access to complete results, including access to quotes from respondents in the comments sections of the survey, and additional discussion of results, are also provided.

2.1 Relevance of Selection Criteria

Three items in the questionnaire regarding the relevance of selection criteria received averages for importance that places them in the lowest quarter of results. In terms of application, the items place in the lower half of results. According to the respondents in the empirical study, the relevance of selection criteria comprised some of the least important issues and was not applied largely. Use of selection criteria established in this study in situations other than the South African context applicable to this study, might make it necessary to reconsider some of them. Many aspects contemplated within the South African context, such as financial considerations would probably remain valid for most situations. Other criteria, e.g. regarding general-purpose programming, or training for industry, would depend on the applicable context. Finally, the relevance of new tendencies in programming would depend on the amount of time that has elapsed since the completion of this study.

2.2 Develop Thinking and Programming Skills

Respondents agree that the selected language should provide an instructional environment that promotes the development of problem solving skills, as the averages for this item distinguish it as both the most important and most applied item. An item with regard to providing students with a firm foundation in good programming practices was applied second most and is the fifth most important. The adequate matching of the adopted programming language to the abilities of students of IT with regard to both level and nature is one of the least important items. Its average for application, however, places it in the top half of items. The latter agrees with literature that the needs, knowledge and abilities of novice users are meaningfully different from those of experienced programmers. Items regarding the development of higher order and critical thinking skills, as well as encouraging a self-regulated approach to solving problems, achieved averages placing them around the top third of items for importance and application. The average for an item that refers to the development of these abilities places it in the middle relative to other criteria with regard to both importance and application. Encouraging programming principles such as programming abstractions [5] and promoting top-down design [6] with step-wise refinement are some of the least important items, but were applied much more, with the latter placing in the top quarter of averages for application.

2.3 Requirements of Appropriate Programming Learning Environments

The item with the third highest average for application refers to providing students with a safe, stable, structured and controlled programming environment [see e.g. 6]. Items with regard to supplying students with understandable error messages, and offering effective debugging tools [1] obtained the fourth and fifth highest averages for application. These items also received averages for importance placing them within the top third of results. Items regarding the programming environment suiting the needs of novice programmers in that it is easy to learn [7], offer relative simplicity of commands [1], and not frustrate students with features for professional programmers [5,6], have averages placing them in the lower half of results. However, these items were applied more than would be expected, with application averages placing them in or close to the top third of results.

2.4 New Tendencies in Programming

The selected programming language offering possibilities for OO design [8] received averages for importance and application placing it in the top quarter of results in both these categories. However, both [2] and [5] point to problems that can occur when using OOP and the difficulties students face when learning to program in an OO style. Using visual languages [1] could offset some of the difficulties with regard to OOP mentioned - this item received averages for importance and application placing this item just inside the lower half of results. Respondents considered the concepts of encapsulation, inheritance and polymorphism some of the least important, with averages placing them in the lowest quarter of results. These items obtained application averages placing them in the lower half of results.

2.5 Issues Influencing Programming Used in Practice

The item regarding the language having reasonable prospects for continued support from its developers [7] has an average that places it as the sixth most important selection criteria, while its average for application is in the higher half of results. Whether or not the language has sufficient capacity for database connectivity also places in the higher half of results for both importance and application. The international standardization of, and trends with regard to, programming languages used [7], place in the lower half of results for importance and in the lowest quarter for application. Considering the popularity and/or demand for specific languages in industry [2] received similar results for importance and application. The language being suitable and having capabilities for the Internet [3] is the least important selection criteria and was also applied second least.

2.6 Affordability, Training and Resources

Respondents considered the affordability of the language to be the second most important item of all. However, the affordability of a selected programming language was applied so little that it places in the lowest quarter of results. The availability of affordable, sufficient in-service teacher training, and having learning and teaching support materials and other resources for the language available to teachers, were considered the third and fourth most important items, but inversely, these items obtained the third and fourth lowest application averages! Similarly, possibilities for training of pre-service teachers [7], and having particularly textbooks available, received averages that place them as the eleventh and seventh most important items. However, application averages for these items are well inside the lowest quarter of results. Although having resources for an OBE approach to teaching the subject [8] has an average placing it in the middle of results, it was in fact applied the least.

2.7 Programming for Various Purposes

Using a general-purpose programming language [7] to support programming for various purposes received averages for both importance and application placing it inside the lower half of results. The same goes for the average for the importance of being able to use the language with academic tools. The average for the application of the latter, as well as the application average for being able to use the language with commercial tools, places these in the lowest quarter of results, while being able to use the language with commercial tools is the fourth least important item.

3 Conclusions

Results in this study arranged selection criteria as being more important relative to others for those with the highest averages, and less important for those obtaining lower averages. During the selection process of a first programming language for high schools, decision makers could pay closer attention to those criteria identified as most important. They could also save time and effort by assigning less importance to criteria with lower averages. Also consider perspectives from literature, especially in cases where these differ from the importance as assigned in this study.

References

1. A.I. Ali and F. Kohun, Suggested Topics for an IS Introductory Course in Java, in: Proceedings of the Informing Science and Information Technology Education Joint Conference (Flagstaff, Arizona, USA, June 2005), pp.33-49, (April 25, 2007); http://proceedings.informingscience.org/InSITE2005/I19f28Ali.pdf

2. Joint Task Force on Computing Curricula, *Chapter 7 Introductory courses*. CC2001 Computer Science volume, Final Report (December 15, 2001); (April 2, 2003); http://www.acm.org/sigcse/cc2001/index.html.

3. J. Barrow, J.H. Gelderblom, and M.G. Miller, *Introducing Delphi programming: theory through practice* (Oxford University Press, Cape Town, 2002).

4. L. Goosen, *Criteria and guidelines for the selection and implementation of a first programming language in high schools* PhD thesis, North-West University (Potchefstroom Campus) (2004); http:// www.puk.ac.za/biblioteek/proefskrifte/2004/goosen_l.pdf

5. N. Mehic and Y. Hasan, Challenges in Teaching Java Technology, in: Proceedings of the 2003 Informing Science Conference (Cracow, Poland, June 2003), pp.301-309, (April 25, 2007); http://proceedings.informing science.org/IS2001Proceedings/pdf/MehicEBKChall.pdf

6. D.B. Palumbo, Programming language/problem-solving research: a review of relevant issues, *Review of educational research* **60**(1), 65-89 (1990).

7. Department of Education, *National Curriculum Statement Grades 10 – 12 (General): Information Technology* (Government Printer, Pretoria, 2003).

8. Department of Education, *National curriculum statement Grades 10-12 (Schools): Guidelines for the development of learning programmes Information Technology* (Extracted from IT.ZIP, August 19, 2003); http://wced.wcape.gov.za/ncs_fet