An Analysis of Criteria for Choosing a First Programming Language in Universities

Yuliia Prokop1[0000-0002-6608-3668], Elena Trofimenko2[0000-0001-7626-0886]
Nikolay Severin1[0000-0002-2706-5205] and Liudmila Bukata1[0000-0003-1707-1633]

[1] O.S. Popov Odessa National Academy of Telecommunications, Odessa, Ukraine
[2] National University "Odessa Academy of Law", Odessa, Ukraine
yulia13.prokop@gmail.com, egt@ukr.net,
n_severin@ukr.net, ygrikluda@gmail.com

Abstract. The choice of the optimal programming language for teaching students of IT specialties the basics of programming at the universities has been the subject of heated debate over the past decades. It is important to focus on modern and significant criteria while choosing a programming language for the introductory course. After analyzing the arguments put forward by teachers in favor of the use of a particular language, as well as the criteria for choosing a language proposed by researchers, a list of the nineteen most frequently called criteria was compiled. Some regional differences in the approaches to the choice of programming languages and the significance of the criteria have been identified. Considering the specifics of the IT sector in Ukraine, based on the study of literature and authors' experience in teaching the basics of programming, a rating of the criteria for choosing the first language was formed. The rating makes it possible to select an optimal programming language, based on the significance of various criteria, taking into account the specific conditions of a particular university.

Keywords: Introductory Programming, First Programming Language, Language Selection Criteria.

1 Introduction

The problem of choosing the first programming language for teaching IT students in universities is very acute nowadays and has caused heated debates over the past decades. Teachers are constantly forced to find a compromise between the motivation of students, their desire to acquire knowledge and skills sufficient to begin practical work as soon as possible, and the need to learn the basic concepts of programming in the form, most accessible for students. The successfully acquired introductory programming course forms the foundation for the effective study of subsequent courses and the mastery of their specialty, similar to the fact that the timeliness of perception of the material and success in studying other subjects depend on mastering the native language in the elementary school. Conversely, the excessive complexity of learning an introductory course can lead to a drop in motivation and low learning outcomes in general.

2 A Variety of Criteria for Choosing a Programming Language for an Introductory Course in Universities

Along with a large number of works in which the authors discuss the advantages and disadvantages of using specific programming languages in the introductory course, great attention is paid in publications to the analysis of the criteria for choosing the first language for instruction.

During this project, we considered the works of authors from different countries, published over the past 15 years. The analysis of works [1-21] revealed significant differences in the choice of criteria of the first studied programming language, and we associate this choice largely with regional features. Different nations have various approaches to building IT education. It is possible that these approaches have a different impact on the choice of the first programming language for introductory courses in universities, justifying their choice by the importance of specific criteria.

The revealed variety of approaches in the initial teaching of programming with the use of a particular language necessitated a comprehensive analysis of the criteria, taking into account their importance in the final choice in a particular university. In addition, in recent years, we can state a change in emphasis in assessing the significance of the criteria for choosing the first language. Thus, since the analyzed researches mainly considered the regional characteristics of various countries and continents, the criteria proposed in them need to be adapted to the situation in Ukraine, in particular.

The purpose of the work is to analyze and rank criteria of the first programming languages for students of IT specialties that can help to identify the optimal programming language for teaching in universities of Ukraine.

3 Formation of the List of Criteria and Their Rating

To accomplish the task, first, we collected and systematized all the criteria identified in publications [1-21] and used by teachers when choosing a language for an introductory programming course. In various papers, the authors formulated and considered a different number of criteria: from one to tens. We took as a basis the criteria proposed and described in the article [16] with the addition of criteria and arguments in favor of the choice of the first educational language encountered in the works of other researchers. We decided to include in the formed list the criteria mentioned in the papers of various researchers more than once. As a result, 19 criteria were identified.

Further, we formed a rating of the criteria for choosing the first programming language depending on the frequency of references in publications. We combined this rating with the rating which we had compiled based on many years of experience teaching the basics of programming in Ukrainian universities, taking into account the traditions of IT education in Ukraine. The most important differences between these two ratings are, in our opinion, the higher value of the criteria for ease of transition to other programming languages and the presence of all the basic programming constructions in the selected language. We adjusted both ratings to a common scale and the average value for each criterion was determined. The result is presented in the form of a diagram in Fig. 1.

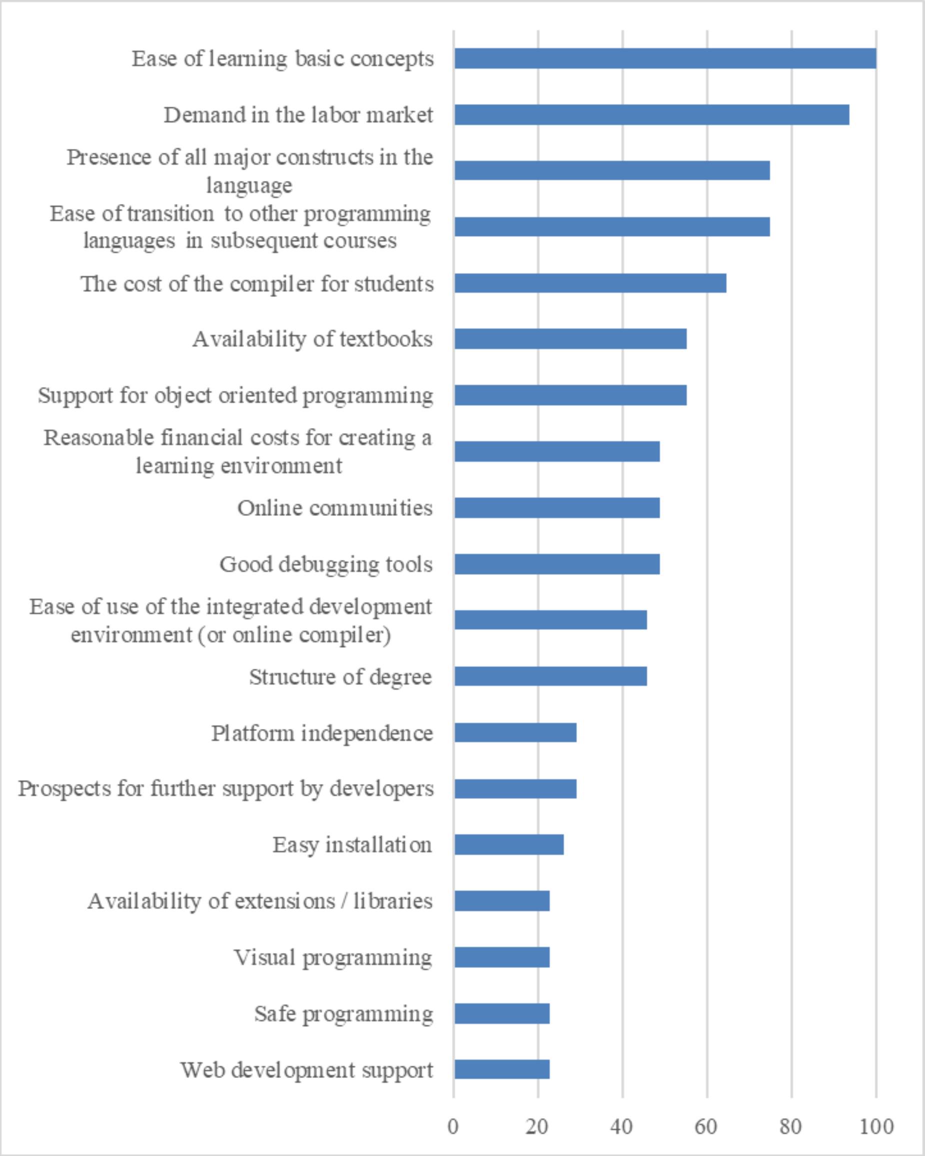

Fig. 1. Rating of the criteria for choosing the first programming language.

Various criteria in the obtained rating have a different degree of objectivity of their values. For example, one can objectively assess the *demand in the labor market* by analyzing the regional labor market. At the same time, the *ease of learning basic concepts* depends on the personal perception of teachers, and on the initial preparation of students, which is very difficult to predict.

4 Analysis of the Criteria for Choosing the First Programming Language

Let us consider the criteria for choosing a programming language. The *ease of learning the basic concepts* is the key to successful mastery of course material and therefore is one of the most significant criteria for choosing a language. Most researchers emphasize the importance of this criterion in their publications.

The most important difference between our rating and one compiled from the literature is the higher value of the *ease of transition to other programming languages* and the *presence of all the basic programming constructions in the language.* In our opinion, the latter is an essential criterion. Traditionally, the introductory course deals with: input-output; branching; loops; one-dimensional and two-dimensional arrays; pointers; strings; structures; file streams. Not all nowadays-common programming languages have basic constructs in their arsenal. For example, C# has no pointers; Java has no pointers and structures [22].

The recent proliferation of dual education puts forward its demands. From the very beginning of their studies, students should learn what they could use in their future activities. This affects the high importance indicators of the *demand in the labor market* and the *ease of transition to other programming languages*. The latter is also relevant for the successful mastering of subsequent courses.

The criterion of the *cost of a compiler for a programming language for students* is significant because Ukrainian students usually use free compilers. A variety of educational materials widely represent all currently popular programming languages on the Internet, thus the criterion of *availability of textbooks* is largely fulfilled. One can say the same about the *ease of use of an integrated development environment* and *good debugging tools* since developers of popular programming languages present many possibilities for their users.

Support for object-oriented programming is not the primary criterion for learning the basics of programming. However, it is of great importance for the further course "Object-oriented programming", since it allows using the same language in both courses. Criterion the *structure of degree* demands the consistency of the use of a given language in the framework of the specialty curriculum. The material taught in subsequent courses should be a logical continuation of what people learned at the beginning.

Criterion the *reasonable financial costs of creating a learning environment* means the costs of retraining lecturers for teaching a new language; expenses for the development of an educational and methodical complex; availability of technical capabilities to install and maintain the necessary software.

A somewhat ambiguous criterion is the existence of *online communities* in which novice programmers can ask for assistance. Sometimes, students receive a ready-made solution and rewrite it, without delving into its essence. In addition, this solution can use constructions not covered by the task, and they confuse the novice even more.

Analysis of the criteria identified in the publications showed that the importance of some of the criteria in them was exaggerated. For example, the presence or absence of *extensions and libraries*, as a rule, cannot have a significant impact on the choice of language. The *supporting web development capabilities* also have a low level of significance, since the basics of programming are usually not related to web development. Due to the availability of online compilers, *platform independence* cannot be the determining factor for choosing a language. Our experience shows that *visual programming* does not affect the effectiveness of the course, but on the contrary, at the initial stage, can make it difficult to understand the educational material and even confuse the novice student.

5 Conclusions

The paper analyzes the criteria and arguments used when choosing a programming language for an introductory programming course at universities for students of IT specialties. We compiled a rating of the nineteen most popular criteria based on the works of different researchers and on our experience of teaching the basics of programming, taking into account Ukrainian IT education specifics.

The composed rating helps to make a choice of a programming language, based on the importance of various criteria inherent in one or another programming language under the prevailing conditions of the course taught. For its practical use, the values obtained can be taken as coefficients for comparative evaluation of various languages.

It can be argued that the illustrated approach and the formed rating can be used not only for the introductory programming course but also for other training courses taking into account the specifics of each of them when teaching IT students in universities.

References

1. Aleksić, V., Ivanović, M.: Introductory Programming Subject in European Higher Education. Informatics in Education, vol. 15, No. 2, pp. 163–82 (2016).
2. Busschots, B.: Teaching Programming – Why the Choice of First Language is Irrelevant, 2008, http://www.bartbusschots.ie/blog/?p=634P, last accessed 2019/1/31.
3. Damyanov, I., Borisova, N.: Programming languages in undergraduate courses and in software industry in Bulgaria. International Journal of Pure and Applied Mathematics, vol. 117, No. 2, 271–278 (2017). DOI: 10.12732/ijpam.v117i2.3.
4. Decker, R., Hirshfield, S.: The top 10 reasons why object-oriented programming can't be taught in CS1. In: Proceeding SIGCSE'94 symposium on Computer science education, pp. 51–55, Phoenix, Arizona, USA (1994).
5. Duffany, J.L.: Choice of Language for an Introduction to Programming Course. In: Excellence in Engineering To Enhance a Country's Productivity, Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014), pp. 1–9. Guayaquil, Ecuador (2014).
6. Goosen, L.: Criteria and guidelines for the selection and implementation of a first programming language in high schools. In: Ph.D. thesis, North-West University (Potchefstroom. Campus) (2004).
7. Ivanoviü, M., Budimac, Z., Radovanoviü, M., Saviü M.: Does the choice of the first programming language influence students' grades? In: International Conference on Computer Systems and Technologies, CompSysTech'15, pp. 305–312, Dublin, Ireland (2015).

8. Jenkins, T.: The First Language – A Case for Python? Innovation in Teaching and Learning in Information and Computer Sciences, vol. 3, issue 2, 1–9 (2004). DOI: 10.11120/ital.2004.03020004/
9. Kruglyk, V., Lvov, M.: Choosing the First Educational Programming Language. In: Proceedings of the 8th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, ICTERI-2012, vol. 848, pp. 188–198, Kherson, Ukraine (2012).
10. Leping, V., Lepp, M., Niitsoo, M., Tõnisson, E., Vene, V., Villems, A.: Python prevails. In: Conference Proceedings of the 2009 International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, CompSysTech 2009, Rousse, Bulgaria (2009). DOI: 10.1145/1731740.1731833.
11. Lvov, M., Kruglyk, V.: Teaching algorithmization and programming using python language. Journal of Information Technologies in Education (ITE), vol. 20, 13–23 (2014). DOI: 10.14308/ite000493.
12. Mason, R., Cooper, G.: Introductory programming courses in Australia and New Zealand in 2013 – trends and reasons. In: Proceedings of the Sixteenth Australasian Computing Education Conference, ACE'14, vol. 148, pp. 139–147, Auckland, New Zealand (2014).
13. Mason, R., Cooper, G., Raadt, M.: Trends in Introductory Programming Courses in Australian Universities – Languages, Environments and Pedagogy. In: Proceedings of the Fourteenth Australasian Computing Education Conference (ACE2012), vol. 123, pp. 33–42, Melbourne, Australia (2012).
14. Merzlikin, P.: Dosvid vikoristannya Python yak pershoyi movi programuvannya dlya studentiv napryamu pidgotovki "Informatika". Naukovi zapiski. Seriya: problemi metodiki fiziko-matematichnoyi i tehnologichnoyi osviti, vol. 2, No 7, 66–69 (2015).
15. Murphy, E., Crick, T., Davenport, J.: An Analysis of Introductory Programming Courses at UK Universities. The Art, Science, and Engineering of Programming, vol. 1, No. 2, article 18, 23 p. (2017). DOI: 10.22152/programming-journal.org/2017/1/18.
16. Parker, K., Ottaway, T., Chao, J.: Criteria for the selection of a programming language for introductory courses. International Journal of Knowledge and Learning, vol. 2, No 1/2, 119–139 (2006). DOI: 10.1504/IJKL.2006.009683.
17. Prokofyeva, N., Uhanova, M., Katalnikova, S., Synytsya, K., Jurenoks, A.: Introductory Programming Training of First Year Students. Procedia Computer Science, vol. 104, pp. 286–293 (2017). DOI: 10.1016/j.procs.2017.01.137.
18. Raadt, M., Watson, R., Toleman, M.: Introductory programming: what's happening today and will there be any students to teach tomorrow? In: Proceedings of the Sixth Australasian Conference on Computing Education, vol. 30, pp. 277–282, Dunedin, New Zealand (2004).
19. Siegfried, R., Chays, D., Herbert, K.: Will there ever be consensus on cs1. In: Proceedings of the 2008 International Conference on Frontiers in Education: Computer Science and Computer Engineering – FECS, vol. 8, pp. 18–23 (2008).
20. Simon, Mason, R., Crick, T., Davenport, J., Murphy, E.: Language Choice in Introductory Programming Courses at Australasian and UK Universities. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE'18, pp. 852–857, Baltimore, Maryland, USA (2018).
21. Zelle, J.M.: Python as a First Language. In: 13th Annual Midwest Computer Conference (MMC 1999). Lisl, IL (1999).
22. Prokop, Y., Trofymenko, O., Kapustin, M.: A study of software development tools that are required in the job market in Ukraine and the world. Proceedings of the O.S. Popov ONAT, vol. 2, 101–108 (2018).