

# CS1: C, JAVA OR PYTHON? TIPS FOR A CONSCIOUS CHOICE

S. Sobral

*Universidade Portucalense (PORTUGAL), REMIT*

## Abstract

Introduction to programming languages is a fundamental point for a student's beginning in the world of computing. Success in programming fundamentals is essential to student success during one's academic career. The matters which are taught in technology courses are a great concern for teachers and course coordinators.

Throughout the last decades it has been verified that the great alterations presented in computer science courses were the new programming languages and paradigms, of at least the transition from structured programming to object oriented programming. The content of curricular units has remained the same. 30 years ago it was common to use a programming language based on BASIC. Twenty years ago Turbo Pascal was commonly used. In the last 10 years has been a discussion about the use of C, java and Python for students who are having their first contact first contact with programming. In previous studies we verified that these are the three programming languages most used in the first semester of computer science courses. In the second semester the most used languages are C, C ++ and java.

There are two paths to follow: using the same programming language in the first two semesters or using one language in the first semester and an alternative one in the second. This last path usually happens with Python and C.

The choice which of programming languages to teach taught is often like that of a football club or religious option; other times it is linked to other important factors, such as applications from employers in the job market.

This article reflects on the choice of which programming language to adopt in CS1. It also lists the languages which are currently most widely adopted in the "real world", and in introductory programming courses in higher education, as well as some studies which help the choice of which programming language to choose. Pedagogical questions and the preparation of students for work are the most important questions addressed by this article, it also list some items that can and should be considered for a conscious choice.

Keywords: Programming languages, CS1.

## 1. INTRODUCTION

Programming is the "visible" skill to acquire in the end of an introductory unit in computer science in higher education. Programming can be considered an art [1], a science [2], a discipline [3] or even the science of abstraction [4].

However using a programming language is no more than a method for the programmer to communicate instructions to the computer. Novice learners usually realize that the focus is on learning the syntax of the programming language, which will lead them to focus on implementation activities rather than activities such as planning, design, or testing. [5]

The following table shows how each of the top ten registered programming languages write the famous "Hello, World!"

*Table 1. Hello World! in ten different programming languages.*

PL	Hello World
C	printf ("Hello World!");
C#	Console.WriteLine("Hello World!");

C++	cout<<"Hello World"
COBOL	display "Hello world!".
Fortran	print *, "Hello world!"
Java	System.out.println("Hello World!");
JavaScript	document.write("Hello world!");
Pascal	writeln ('Hello World!');
Python	print "Hello, World!"
Visual Basic .NET	Console.WriteLine("Hello world!")

Each of the ten programming languages presented in the previous table have a different notation, however quite similar to a basic algorithm like "Hello World!".

For some programming can be very difficult [6] [7], while for others it may be easy [8]. We have no doubt:

Success is achieved through a good deal of study, research, planning, persistence and preferably a passion for the activity.

## 2. PROGRAMMING LANGUAGES IN THE REAL WORLD, CURRENTLY

There are several reasons why thousands of high-level programming languages exist and new ones continue to emerge [9] for example:

- Evolution: The late 1960s and early 1970s saw a revolution in "structured programming," in which the GoTo-based flow control of languages such as Fortran, Cobol, and Basic gave way to while loops, case statements (switch). In the late 1980s, Algol, Pascal and Ada began to give way to the object-oriented language structure like Smalltalk, C ++, Eiffel.
- Special Purposes: Some are designed for a specific purpose. C is good for low level system programming. Prolog is good for reasoning about logical relationships between data. Each can be successfully used for a variety range of tasks, but the emphasis of a program is clearly on the respective specialty.
- Personal preference: Different people like different things. Some people love C's conciseness while others hate it, for example.

According to Stack Overflow Annual Developer Survey [10], with over 90,000 answers to over 170 countries, by 2019 the most widely used programming language is Javascript (Table 2).

*Table 2. Top15 Most Used Programming Languages in 2019 [10].*

PL	%
JavaScript	67.8%
HTML/CSS	63.5%
SQL	54.4%
Python	41.7%
Java	41.1%
Bash/Shell/PowerShell	36.6%
C#	31.0%
PHP	26.4%
C++	23.5%
TypeScript	21.2%
C	20.6%
Ruby	8.4%
Go	8.2%
Assembly	6.7%

Swift	6.6%
-------	------

In September 2019, TIOBE Programming Community index [11], the programming language popularity indicator, introduced Java as the most popular programming language (Table 3), followed by C and Python.

Table 3. Top7, Indicator of popularity of programming languages, TIOBE [11].

PL	Ratings
Java	16,66%
C	15,21%
Python	9,87%
C++	5,64%
C#	3,40%
Visual Basic .NET	3,29%
JavaScript	2,13%

The technology of the most searched electronic sites (Table 4) according to Wikipedia<sup>1</sup> [12] is also varied in the use of back-end languages; However, Javascript is almost always used at the front end.

Table 4. The technology of the most searched websites [12].

WebSite	back-end language
Amazon.com	Java, C++, Perl
Bing	C++, C#
eBay.com	Java, JavaScript, Scala
Facebook.com	Hack, PHP (HHVM), Python, C++, Java, Erlang, D, XHP, Haskell
Google.com	C, C++, Go, Java, Python
Linkedin.com	Java, JavaScript, Scala
MSN.com	C#
Pinterest	Python (Django), Erlang
Twitter.com	C++, Java, Scala, Ruby
Wikipedia.org	PHP, Hack
WordPress.com	PHP
Yahoo	PHP
YouTube.com	C, C++, Python, Java, Go

### 3. PROGRAMMING LANGUAGES IN CS1, CURRENTLY

In Portugal [13], in the 2016-2017 school year, the most common first-year programming language sequence in 46 courses analysed was just C (48%), followed by Java (22%), C and Haskell (9%), C and Java (4%), Scheme and Java (4%). There were also residual sequences of Excel and C, just Python, Python, HTML and Java, Python and Java, Schem and C ++ and XML and Java. Regarding the 10 Portuguese first cycle courses in Computer Engineering considered

<sup>1</sup> Wikipedia isn't a trustworthy source of information due its collaborative characteristics!

the most significant [14], the most common sequences found were only Java or Python and C (both with 30%), only C (20%), Python and Java or Haskell and C (both with 10%).

According to the document “An Analysis of Introductory Programming Courses at UK Universities” [15]:

- The vast majority of courses surveyed (59 of 80, 73.8%) use only one programming language; 17 (21%) reported using two.
- The most commonly used language is Java (46%), followed by the C "family" (C, C ++ and C #) (23.6%) and Python (13.2%). Javascript and Haskell are much less used.
- 82.7% of those who use Java justified it with the fact that is object oriented.
- 72.7% of those using Python refer the pedagogical benefits as reason for their choice.

According to the document “Introductory Programming Courses in Australasia in 2016” [16], regarding Universities of Australia and New Zealand:

- Of the 48 courses studied, 15 used Java, 15 Python, 8 C, 5 C #, 2 Visual Basic and 2 Processing.
- The reasons given for choosing Python and Java are quite different: Python is adopted for its pedagogical benefits (67%), availability / cost (53%) and platform independence (40%). On the other hand, Java credited for its industry-relevant (92%), object-oriented (86%) and platform independence (62%).

“What language? - The choice of an introductory programming language” [17], a study regarding 496 four-year courses in the United States found that Java was used by 41.94%, Python 26.45%, C ++ 19.35%, C 4.52%, C # 0.65%. And the reasons behind it's choice were (multiple choice option): Programming language features 26.19%, Ease of learning 18.81%, Job opportunities for students 14.76%, Popularity 13.10%, Institutional tradition 8.57%, choice of advisory board 5.95%, availability of teachers or scheduling restrictions 5%.

A study [18] of 152 CS1 units from different countries concluded that Java is by far the most common CS1 language, it is used in 74 (49%) of the 152 programs. The second most frequent is Python, with 36 (24%). C ++ comes in 30 (20%) followed by C in 8 (5%).

Today, with few exceptions, the academy follows the “real world”: the C family (C, C ++, C #), Python, Java, and JavaScript are undoubtedly the programming languages more adopted in introductory programming units.

#### **4. CHOICE OF INITIAL PROGRAMMING LANGUAGE**

In curriculum recommendations [19] [20] [21] [22] [23] [24] [25] [26] there is no indication on which particular programming language to choose from. However, it is always said that they should have the simplest possible usability and syntax for better learning. Language choice has always been a matter of concern [27] [28] [29] [30] [31] [32] [33] [34] [35] [35] [36] [37] [38] [39] [40] [41] [31].

Dijkstra [42] called attention to the important problem on which initial programming language to choose: “The tools we are trying to use and the language or notation we are using to express or record our thoughts, are the major factors determining what we can think or express at all! The analysis of the influence that programming languages have on the thinking habits of its users, and the recognition that, by now, brainpower is by far our scarcest resource, they together give us a new collection of yardsticks for comparing the relative merits of various programming languages.”.

The programming languages chosen for introductory programming courses often seems like a religious or football issue. In reflection-teaser “The Programming Language Wars” [43] it is even said that “Programming language wars are a major social problem causing serious problems in our discipline” leading to “massively duplicating efforts” and “reinventing the wheel constantly.” Choosing the best programming language is often an emotional issue, leading to major debates [44] but for Guerreiro [45] “It is up to us to have an open, exploratory attitude and at the same time not dogmatically accept what those who make the most noise say. In fact, I think that we

should also pass this on to students in order to help them develop their critical thinking, and to be able, sooner or later, to choose the languages and tools that can best respond to their needs. Programming languages are the fundamental basis of programming, but trends change dramatically over time. Professionals will not use the same programming language, or even the same programming model, for their entire professional career”.

In the past, several attempts have been made to sort programming languages as the first language. [46]. There are numerous comparisons between the most commonly used languages: like Python vs C++ [47], Python vs C [48], Java vs Python [49], C++ vs. Java [50]. Any of the three / four most commonly used languages today is free, has good support and a large user community and is reliable and efficient.

The ease of learning can be discussed: C will have a more complicated syntax than Python. The major differences are the use of pointers (C only), parameter passing by reference and value (C only), programming paradigm (procedural in C, object oriented in others), being compiled or interpreted (C and Python/Java respectively).

In fact, two of the most important points are pedagogical issues and student preparation for the world of work. Students are often more motivated to study a familiar language known to employers and not only for educational purposes only.

We consider several items:

- Student learning: “programming in small”; Easy learning of fundamental concepts, advanced features for subsequent programming courses, more or less complicated, suitable for learning, promotes correct writing, good support and good teaching material.
- Language features: Be reliable and efficient, have rigor, speed, usability, ease of portability, reuse, good development environment, debugging facilities, secure Code, good user community and preferably be Open Source.
- Institutional: reasonable financial cost, availability of Academic / Student Version, Academic Acceptance, Availability of textbooks, course objectives, Teacher preferences, and relationship to other course units.

For the right choice we have to make a weighted average of each of these items by importance and evaluate each of the possible programming languages.

## **5. CONCLUSIONS**

There are many programming languages that are adopted for different reasons: a matter of evolution, purpose of use or even personal taste.

The choice of the programming language for introductory teaching must accompany evolution, but because it has a propaedeutic character. The choice must meet several requirements, namely regarding pedagogical criteria and acceptance from the “real world”.

This article lists a set of items that can be used to make a conscious choice.

There isn't, and probably never will be, consensus as to which language should be chosen to introduce the student to the world of computer science.

However the first programming language of a computer future is no more than the beginning of a long walk.

## REFERENCES

- [1] D. Knuth, *The Art of Computer Programming*, Addison-Wesley, 1968.
- [2] D. Gries, *The science of programming*, Springer, 1981.
- [3] E. W. Dijkstra, *A Discipline of Programming*, Prentice Hall, 1976.
- [4] A. Aho and J. D. Ullman, *Foundations of Computer Science: C Edition (Principles of Computer Science Series)*, W. H. Freeman, 1994.
- [5] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B.-D. Kolikant, C. Laxer, L. Thomas, I. Utting and T. Wilusz, "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students," in *ITiCSE on Innovation and technology in computer science education*, 2001.
- [6] S. Bergin and R. Reilly, "Programming: Factors that Influence SuccessSusan," in *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, 2005.
- [7] B. A. Becker, G. Glanville, R. Iwashima, C. McDonnell, K. Goslin and C. Mooney, "Effective Compiler Error Message Enhancement for Novice Programming Students," *Computer Science Education*, pp. 148-175, 2016.
- [8] A. Luxton-Reilly, "Learning to Program is Easy," in *ACM Conference on Innovation and Technology in Computer Science Education*, 2016.
- [9] M. L. Scott, *Programming Language Pragmatics*, 3rd edition ed., Elsevier, 2009.
- [10] Stackoverflow.com, "Stackoverflow," 2019. [Online]. Available: <https://insights.stackoverflow.com/survey/2019>.
- [11] TIOBE Software BV, "TIOBE," Set. 2019. [Online]. Available: <https://www.tiobe.com/tiobe-index/>.
- [12] Wikipedia, "Programming languages used in most popular websites," Setembro 2019. [Online]. Available: [https://en.wikipedia.org/wiki/Programming\\_languages\\_used\\_in\\_most\\_popular\\_websites](https://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites).
- [13] S. R. Sobral, "Bachelor's and master's degrees integrated in Portugal in the area of computing: a global vision with emphasis on programming UCS and programming languages used," in *11th annual International Conference of Education, Research and Innovation*, 2018.
- [14] S. R. Sobral, "Introduction to programming: Portrait of Higher Education in computer science in Portugal," in *11th International Conference on Education and New Learning Technologies*, 2019.
- [15] E. Murphy<sup>1</sup>, T. Crick<sup>2</sup> and J. H. Davenport, "An Analysis of Introductory Programming Courses at UK Universities," *The Art, Science, and Engineering of Programming*, vol. 1, no. 2, 2017.

- [16] R. Mason and Simon, "Introductory Programming Courses in Australasia in 2016," in *Nineteenth Australasian Computing Education Conference*, 2017.
- [17] O. Ezenwoye, "What language? - The choice of an introductory programming language," *48th Frontiers in Education Conference, FIE 2018*, 2018.
- [18] B. A. Becker and T. Fitzpatrick, "What Do CS1 Syllabi Reveal About Our Expectations of Introductory Programming Students?," in *50th ACM Technical Symposium on Computer Science Education*, 2019.
- [19] W. F. Atchison, S. D. Conte, J. W. Hamblen, T. E. Hull, T. A. Keenan, W. B. Kehl, E. J. McCluskey, S. O. Navarro, W. C. Rheinboldt, E. J. Schweppe, W. Viavant and D. M. Young, Jr., "Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science," *Communications of the ACM*, vol. v.11 n.3, pp. 151-197, Março 1968.
- [20] R. H. Austing, B. H. Barnes, D. T. Bonnette, G. L. Engel and G. Stokes, "Curriculum '78: recommendations for the undergraduate program in computer science— a report of the ACM curriculum committee on computer science," *Communications of the ACM*, vol. v.22 n.3, pp. 147-166, Março 1979.
- [21] E. B. Koffman, P. L. Miller and C. E. Wardle, "Recommended curriculum for CS1, 1984," *Communications of the ACM*, vol. v.27 n.10, pp. .998-1001, Outubro 1984.
- [22] P. J. Denning, "Computing as a discipline," *Communications of the ACM*, vol. 32, no. 1, pp. 9-23, 1989.
- [23] A. B. Tucker and ACM/IEEE-CS Joint Curriculum Task Force., Computing curricula 1991 : report of the ACM/IEEE-CS Joint Curriculum Task Force, ACM Press, 1991, p. 154.
- [24] The Joint Task Force IEEE and ACM, "CC2001 Computer Science, Final Report," 2001.
- [25] L. Cassel, A. Clements, G. Davies, M. Guzdial and R. McCauley, "Computer Science Curriculum 2008: An Interim Revision of CS 2001," ACM, 2008.
- [26] Task force ACM e IEEE, "Computer Science Curricula 2013," ACM and the IEEE Computer Society, 2013.
- [27] C. Smith and J. Rickman, "Selecting Languages for Pedagogical Tools in the Computer Science Curriculum," in *Proceedings of the sixth SIGCSE technical symposium on Computer science education*, 1976.
- [28] R. L. Wexelblat, "Discussion), First programming language: Consequences (Panel," in *Discussion), First programming language: Consequences (Panel*, 1979.
- [29] A. L. Tharp, "Selecting the "right" programming language," in *SIGCSE '82 technical symposium on Computer science education*, Indianapolis, Indiana, USA, 1982.
- [30] R. Duke, E. Salzman, J. Burmeister, J. Poon and L. Murray, "Teaching programming to beginners - choosing the language is just the first step," in *ACSE '00 Proceedings of the Australasian conference on Computing education*, 2000.
- [31] L. Mannila and M. d. Raadt, "An objective comparison of languages for teaching introductory programming," in *6th Baltic Sea conference on Computing education research: Koli Calling 2006*, 2006.

- [32] E. Roberts, "The Dream of a Common Language: The Search for Simplicity and Stability in Computer Science Education," in *35th SIGCSE technical symposium on Computer science education*, 2004.
- [33] K. Parker and B. Davey, "The History of Computer Language Selection," *IFIP Advances in Information and Communication Technology*, pp. 166-179, 2012.
- [34] M. Laakso, E. Kaila, T. Rajala and T. Salakoski, "Define and Visualize Your First Programming Language," in *8th IEEE International Conference on Advanced Learning*, 2008.
- [35] C. S. Collberg, "Data structures, algorithms, and software engineering," in *Software Engineering Education - SEI Conference 1989*, 1989.
- [36] K. Bruce, S. N. Freund, R. Harper, J. Larus and G. Leavens, "What a Programming Languages Curriculum Should Include," in *SIGPLAN Workshop on Undergraduate Programming Language Curricula*, 2008.
- [37] K. N. King, "The evolution of the programming languages course," *ACM SIGCSE Bulletin*, vol. 24, no. 1, pp. 213-219, 1992.
- [38] J. Howatt, "A project-based approach to programming language evaluation," *ACM SIGPLAN Notices*, vol. 30, no. 7, pp. 37-40, 1995.
- [39] P. A. Luker, "Never mind the language, what about the paradigm?," in *twentieth SIGCSE technical symposium on Computer science education*, 1989.
- [40] T. A. Budd and R. K. Pandey, "Never Mind the Paradigm, What About Multiparadigm Languages?," *ACM SIGCSE Bulletin*, vol. 27, no. 2, pp. 25-30, 1995.
- [41] K. R. Parker, J. T. Chao, T. A. Ottaway and J. Chang, "A Formal Language Selection Process," *Journal of Information Technology Education*, vol. 5, no. 1, pp. 133-151, 2006.
- [42] E. W. Dijkstra, "The Humble Programmer," *Communications of ACM*, vol. 15, no. 10, 1972.
- [43] A. Stefik and S. Hanenberg, "The Programming Language Wars: Questions and Responsibilities for the Programming Language Community," in *2014 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software*, 2014.
- [44] L. Gosen, "A Brief History of Choosing First Programming languages," *History of Computing and Education*, vol. 3, 2008.
- [45] P. Guerreiro, "A mesma velha questão: como ensinar Programação?," in *Quinto Congresso Iberoamericano de Educación Superior*, 1986.
- [46] K. S. A. F. I. S. A. A. Farooq MS, "An Evaluation Framework and Comparative Analysis of the Widely Used First Programming Languages," *PLoS ONE*, 2014.
- [47] N. Alzahrani, F. Vahid, A. Edgcomb, K. Nguyen and R. Lysecky, "Python Versus C++: An Analysis of Student Struggle on Small Coding Exercises in Introductory Programming Courses," in *49th ACM Technical Symposium on Computer Science Education*, 2018.
- [48] J. Wainer and E. Xavier, "A Controlled Experiment on Python vs C for an Introductory Programming Course: Students' Outcomes," *ACM Transactions on Computing Education*, vol. 18, no. 3, 2018.



- [49] K. McMaster, S. Sambasivam, R. Rague and S. Wolthuis, "Java vs. Python Coverage of Introductory Programming Concepts: a Textbook Analysis," *Information Systems Education Journal*, vol. 15, no. 3, pp. 4-13, 2017.
- [50] W. Farag, S. Ali and D. Deb, "Does language choice influence the effectiveness of online introductory programming courses?," in *14th annual ACM SIGITE conference on Information technology education*, 2013.
- [51] S. R. Sobral, B-learning em disciplinas introdutórias de programação, Universidade do Minho, 2008.
- [52] S. R. Sobral and P. Pimenta, "O ensino da programação: exercitar a distancia para combate às dificuldades," in *4ª Conferência Ibérica de Sistemas e Tecnologias de Informação*, 2009.
- [53] J. R. Lima, Programação de computadores, Porto Editora, 1991.
- [54] J. R. Rice and S. Rosen, "History of the Computer Sciences Department at Purdue University," Department of Computer Science, Purdue University, 1990.
- [55] E. Giangrande Jr., "CS1 programming language options," *Journal of Computing Sciences in Colleges*, vol. v22, no. 3, pp. 153-160, 2007.
- [56] J. G. Kemeny and T. E. Kurtz, BASIC - A Manual for BASIC, the elementary algebraic language, Dartmouth College, 1964.
- [57] E. W. Dijkstra, "Go To Statement Considered Harmful," *Communications of the ACM*, vol. 11, no. 3, pp. 147-148, 1968.
- [58] D. Knuth, "Structured Programming with go to Statements," *Computing Surveys*, vol. 6, no. 4, pp. 261-301, 1974.
- [59] O. Dahl, E. Dijkstra and C. Hoare, Structured programming, Academic Press Ltd, 1972.
- [60] N. Wirth, "The Programming Language Pascal," in *Pioneers and Their Contributions to Software Engineering*, Springer, 1971.
- [61] D. Gupta, "What is a good first programming language?," *Crossroads, The ACM Magazine for Students*, vol. 10, no. 4, 2004.
- [62] S. Levy, "Computer language usage in CS1: survey results," *ACM SIGCSE Bulletin*, vol. 7, no. 3, pp. 21-26, 1995.
- [63] R. McCauley and B. Manaris, "Computer science degree programs: what do they look like? A report on the annual survey of accredited programs," *ACM SIGCSE Bulletin*, vol. 30, no. 1, pp. 15-19, 1998.
- [64] S. R. Sobral, "30 YEARS OF CS1: PROGRAMMING LANGUAGES EVOLUTION," in *12th annual International Conference of Education, Research and Innovation*, 2019.
- [65] G. D. M. N. S. J. Siegfried RM, "A Longitudinal Analysis of the Reid List of First," *Information Systems Education Journal*, vol. 10, no. 4, pp. 47-54, 2016.
- [66] V. ALEKSIĆ and M. IVANOVIĆ, "Introductory Programming Subject in European Higher Education," *Informatics in Education*, vol. 15, no. 2, p. 163-182, 2016.

