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 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>
<thead>
<tr>
<th>PL</th>
<th>Hello World</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>printf(&quot;Hello World!&quot;);</td>
</tr>
<tr>
<td>C#</td>
<td>Console.WriteLine(&quot;Hello World!&quot;);</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>PL</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript</td>
<td>67.8%</td>
</tr>
<tr>
<td>HTML/CSS</td>
<td>63.5%</td>
</tr>
<tr>
<td>SQL</td>
<td>54.4%</td>
</tr>
<tr>
<td>Python</td>
<td>41.7%</td>
</tr>
<tr>
<td>Java</td>
<td>41.1%</td>
</tr>
<tr>
<td>Bash/Shell/PowerShell</td>
<td>36.6%</td>
</tr>
<tr>
<td>C#</td>
<td>31.0%</td>
</tr>
<tr>
<td>PHP</td>
<td>26.4%</td>
</tr>
<tr>
<td>C++</td>
<td>23.5%</td>
</tr>
<tr>
<td>TypeScript</td>
<td>21.2%</td>
</tr>
<tr>
<td>C</td>
<td>20.6%</td>
</tr>
<tr>
<td>Ruby</td>
<td>8.4%</td>
</tr>
<tr>
<td>Go</td>
<td>8.2%</td>
</tr>
<tr>
<td>Assembly</td>
<td>6.7%</td>
</tr>
</tbody>
</table>
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].**

<table>
<thead>
<tr>
<th>PL</th>
<th>Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>16.66%</td>
</tr>
<tr>
<td>C</td>
<td>15.21%</td>
</tr>
<tr>
<td>Python</td>
<td>9.87%</td>
</tr>
<tr>
<td>C++</td>
<td>5.64%</td>
</tr>
<tr>
<td>C#</td>
<td>3.40%</td>
</tr>
<tr>
<td>Visual Basic .NET</td>
<td>3.29%</td>
</tr>
<tr>
<td>JavaScript</td>
<td>2.13%</td>
</tr>
</tbody>
</table>

The technology of the most searched electronic sites (Table 4) according to Wikipedia¹ [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].**

<table>
<thead>
<tr>
<th>WebSite</th>
<th>back-end language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon.com</td>
<td>Java, C++, Perl</td>
</tr>
<tr>
<td>Bing</td>
<td>C++, C#</td>
</tr>
<tr>
<td>eBay.com</td>
<td>Java, JavaScript, Scala</td>
</tr>
<tr>
<td>Facebook.com</td>
<td>Hack, PHP (HHVM), Python, C++, Java, Erlang, D, XHP, Haskell</td>
</tr>
<tr>
<td>Google.com</td>
<td>C, C++, Go, Java, Python</td>
</tr>
<tr>
<td>LinkedIn.com</td>
<td>Java, JavaScript, Scala</td>
</tr>
<tr>
<td>MSN.com</td>
<td>C#</td>
</tr>
<tr>
<td>Pinterest</td>
<td>Python (Django), Erlang</td>
</tr>
<tr>
<td>Twitter.com</td>
<td>C++, Java, Scala, Ruby</td>
</tr>
<tr>
<td>Wikipedia.org</td>
<td>PHP, Hack</td>
</tr>
<tr>
<td>WordPress.com</td>
<td>PHP</td>
</tr>
<tr>
<td>Yahoo</td>
<td>PHP</td>
</tr>
<tr>
<td>YouTube.com</td>
<td>C, C++, Python, Java, Go</td>
</tr>
</tbody>
</table>

³ Wikipedia isn’t a trustworthy source of information due its collaborative characteristics!

---

³ 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] [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


[54] J. R. Rice and S. Rosen, "History of the Computer Sciences Department at Purdue University," Department of Computer Science, Purdue University, 1990.


