



IJSRIS
journal
ISSN: 2820-7157

★ INTERNATIONAL JOURNAL OF ★

**SCIENTIFIC RESEARCH AND
INNOVATIVE STUDIES**

Advancing Knowledge, Inspiring Innovation

 www.ijsrisjournal.com



**INDEXED
JOURNAL**

 **PEER REVIEWED**
Rigorous & Transparent

 **DOI: CROSSREF**
Member

 **OPEN ACCESS**
For Global Visibility

 **BI-MONTHLY**
Timely Publication

 **WIDE READERSHIP**
High Impact & Visibility

 **AUTHOR FRIENDLY**
Fast | Supportive | Reliable



RECEIVED
17 August 2026



ACCEPTED
09 September 2026



PUBLISHED
1 October 2026



October 2026



Volume 5



Number 5



Pages from:
311 to 318

Analysis of Programming Languages and Teaching Methodologies in the First-Year Computer Science Degree Programme in Lubumbashi

MANDELA NSENGIYUMBA David¹

¹PhD student at the National Pedagogical University, Department Secretary for Computer Science in charge of Tutoring/Research at the Higher Pedagogical Institute of Lubumbashi, DRC, mandela.dav33@gmail.com , ORCID 0009-0003-2238-3751

LICENSE



This article is licensed under the
Creative Commons Attribution
4.0 International License (CC BY 4.0).

 <https://creativecommons.org/licenses/by/4.0/>



PREFIX :
10.63883

IJSRIS JOURNAL

OPEN ACCESS



This is an open access article
distributed under the terms of
open access.

ABSTRACT

This article examines the dynamics of teaching programming languages in Lubumbashi, focusing on identifying the languages taught and the chosen teaching methodologies. Using a descriptive survey method, data were collected from 514 students and 12 lecturers across 10 higher education

institutions. The results reveal a strong reliance on traditional languages such as C and Visual Basic 6.0, despite students' preferences for more modern tools such as Java or Python. Furthermore, the study highlights a predominance of interrogative and demonstrative methods, whilst modern active learning strategies remain largely underutilised. The findings suggest a disconnect between the national curriculum

and global technological trends, calling for an urgent update to teaching practices.

Keywords: Programming language, Teaching methodology, Didactics, Higher education, Lubumbashi.

I. Introduction

The teaching of computer science, and more specifically programming, lies at the heart of the training of future technology professionals in the Democratic Republic of the Congo. In Lubumbashi, although demand for training is constantly growing, the teaching of programming faces major didactic challenges. As Hartmann et al. (2006) point out, computer science cannot be taught haphazardly, but requires sound didactic reflection to adapt content to technological developments.

However, the Congolese education system appears to suffer from ‘pedagogical obsolescence’ due to a static national curriculum dating from 2004, which specifies neither the languages to be prioritised nor the precise skills to be acquired. This study aims to identify the languages actually taught in the first year of the undergraduate degree and to analyse the methodologies chosen by teachers to convey this complex knowledge.

Research question

To what extent does the discrepancy between the national curriculum, the lack of pedagogical training for teachers and the continued use of traditional programming languages hinder the acquisition of modern software skills among first-year computer science students in Lubumbashi?

Objective

To assess the impact of choices regarding programming languages and teaching methodologies on the quality of first-year computer science education in Lubumbashi, with a view to proposing guidelines for a revision of the curriculum and teaching methods adapted to current technological standards.

Hypothesis

We believe that the choice of programming languages and teaching methods in the first-cycle computer science programme in Lubumbashi is primarily dictated by the perpetuation of traditional practices and the obsolescence of the national curriculum, rather than by the needs of the labour market and innovations in computer science pedagogy.

II. Methodology

This research adopts a descriptive approach based on **survey methods** and **literature review**.

- **Sampling:** The study involved a sample of **514 students** (36.53 per cent of the population) and **12 lecturers** from **10 institutions** (public, private and religious) in Lubumbashi, such as the ISP, the ISS, the UPL and the UML.
- **Data collection:** Two structured questionnaires were administered to gather views on the languages taught, technological preferences and teaching methods.
- **Analysis:** The data were statistically analysed using the **R software** and chi-squared tests to verify the significance of the results.

III. Results

1. Survey of programming languages

1.1. For teachers

a. Programming language taught

In order to identify the programming language taught in the first year of the undergraduate degree programme according to the institution’s curriculum, the following question was asked: ‘Which programming language do you teach in the first year of the Computer Science programme according to your institution’s curriculum?’. The responses provided by our respondents are presented in the table below:

Table 1: Programming language taught according to the institution’s curriculum.

Programming language taught	Number of students	%
C++, VB.6.0	1	8.33
VBNET	1	8.33
C	2	16.67
C, C++	1	8.33

C, Python	1	8.33
C, HTML	1	8.33
C, JAVA, PYTHON	1	8.33
C++	1	8.33
C++, Python	1	8.33
JAVA	1	8.33
PASCAL	1	8.33
Total	12	100.00

Analysis of Table 1 shows that out of a total of 12 teachers representing 100 per cent; only 2, or 16.67 per cent, of the teachers teach the C programming language. The remaining teachers each account for 8.33 per cent of those teaching C++ and VB6.0, or C and C++, or C and Python, or C, Java and Python, or C++ and Python, or C++, Java or Pascal.

The chi-squared test yielded a value of 0.833 ($p = 1$) with 10 degrees of freedom; this leads us to conclude that the difference is not significant.

b. Programming language preferred by the teacher

With the same aim in mind, we sought to determine which programming language the teacher would have preferred to teach in the first year of the undergraduate degree. The following question was therefore posed: 'Based on your own experience, which programming language would you have preferred to teach in the first year of the Computer Science degree?'. The responses provided by our respondents are presented in the table below:

Table 2: Programming languages preferred by teachers.

Preferred programming language	Number of students	%
C++, Python, VB.NET	1	8.33
Python, VB.NET	1	8.33
C	1	8.33
C, C++	1	8.33
C, C++, Java	1	8.33
C, C++, Python	1	8.33
C++	1	8.33
C++, Python	1	8.33
C#, Java, Python	1	8.33
JAVA	1	8.33
VB.NET	2	16.67
Total	12	100.00

Analysis of Table 2 shows that, out of a total of 12 teachers representing 100 per cent, only 2, or 16.67 per cent, of the teachers would have liked to teach VB.NET. The remaining teachers, each accounting for 8.33 per cent, would have liked to teach C++, Python and VB.NET; or C and C++; or C++ and Python; or C, C++ and Python; or C++ and Python; or C, C++ and Java; or C++; or Java.

The chi-squared test yielded a value of 0.833 ($p = 1$) with 10 degrees of freedom; this leads us to conclude that the difference is not statistically significant.

c. Reason for the teacher's choice of language

Following the same line of reasoning, we sought to understand why the teacher might have chosen to teach one particular

programming language rather than another. The following question was therefore posed: ‘Why did you opt for the language mentioned above?’. The responses provided by our respondents are presented in the table below:

Table 3: Reasons for the teacher’s choice of language.

Reason for choice	Number of pupils	%
Ranked amongst the best languages	5	41.67
In-demand in the job market	1	8.33
Facilitates learning	4	33.33
It’s topical	2	16.67
Total	12	100.00

The results in Table 3 show that, out of a total of 12 teachers representing 100 per cent, the majority – comprising 5 teachers, or 41.67 per cent – chose a programming language because it is ranked amongst the best languages. This was followed by 4 teachers (33.33 per cent) who chose a language because it is ranked amongst the best languages. Next, 16.67 per cent (2 teachers) would have liked to teach a language because it is topical. Finally, only one teacher out of 12 (8.33 per cent) would have liked to teach a language because it is in demand in the world of work.

The chi-squared test yielded a value of 3.333 ($p = 0.343$) with three degrees of freedom; this leads us to conclude that the difference is not statistically significant.

1.2. For Students

a. Programming languages learnt

In order to identify the programming language taught in the first year of the degree programme, as set out in the institution’s curriculum, the following question was asked: ‘Which programming language did you learn in your first year of Computer Science, as set out in your institution’s curriculum?’. The responses provided by our respondents are presented in the table below:

Table 4: Programming language learnt/student.

Programming language learnt	Number of students	%
C, C++	150	29.18
C, HTML	23	4.47
C, JAVA, PYTHON	8	1.56
C, Python	7	1.36
C, VB.NET	37	7.20
C, VB6.0	178	34.63
C++, VB6.0	13	2.53
C++, Python	5	0.97
JAVA	88	17.12
PASCAL, COBOL	5	0.97
Total	514	100.00

Analysis of Table 4 shows that out of a total of 514 students representing 100 per cent, only 178 (34.63 per cent) stated that they had learnt the C and VB6.0 programming languages, followed by 150 students (29.18 per cent) who stated that they had learnt C and C++; subsequently, 88 students, or 17.12 per cent, stated that they had learnt Java, followed by 37 students, or 7.20 per cent, who stated that they had learnt C and VB.NET, then 23 students (4.47 per cent) who stated they had learnt C and HTML, followed by 13 students (2.53 per cent) who confirmed they had learnt C++ and VB6.0, followed by 8 students (1.56 per cent) who confirmed they had learnt C, Java and Python, followed by 7 students (1.36 per cent) who confirmed they had learnt the C and Python languages, and finally, 0.97 students (5) who jointly confirmed they had learnt the Pascal and COBOL languages or C++ and Python.

The chi-squared test yielded a value of 734.21 ($p = 0.000$) with 9 degrees of freedom; this leads us to reject the null hypothesis.

b. Programming languages preferred by students

With the same aim in mind, we sought to specify which programming language the students would have liked to learn in their first year of the degree programme. The following question was therefore asked: ‘Which programming language would you have liked to learn in your first year of computer science?’. The responses provided by our respondents are presented in the table below:

Table 5: Programming languages preferred by students.

Preferred programming language	Number of students	%
C	89	17.32
C++	76	14.79
C++, PHP	1	0.19
C++, Python	1	0.19
C#	19	3.70
C#, PHP	1	0.19
HTML	6	1.17
JAVA	89	17.32
JAVA, PHP	2	0.39
Java, Python	7	1.36
JAVA, VB.NET	2	0.39
JAVASCRIPT	42	8.17
PASCAL	1	0.19
PHP	25	4.86

PYTHON	70	13.62
R	3	0.58
VB.NET	80	15.56
Total	514	100.00

Analysis of Table 5 shows that out of a total of 514 students, representing 100 per cent, only 89 (17.32 per cent) would have liked to learn either the C or Java programming language, followed by 80 students (15.56 per cent) who would have liked to learn VB.NET, then 76 students (14.79 per cent) who would have liked to learn C++ and PHP, followed by 70 students (13.62 per cent) who would have liked to learn Python. Finally, the remaining students’ choices ranged from 0.19 per cent (1 student) to 8.17 per cent (42 students) for the following languages: JavaScript, C#, R, Pascal, Java and Python, HTML, or C++ and Python.

The chi-squared test yielded a value of 669.19 ($p = 0.000$) with 16 degrees of freedom; this leads us to reject the null hypothesis.

c. Reasons for students’ choice of language

Following the same line of reasoning, we sought to understand why students chose to learn one programming language rather than another. The following question was therefore asked: ‘Why did you choose the language listed above?’. The responses provided by our respondents are presented in the table below:

Table 6: Reasons for students’ choice of programming language.

Reason for choice	Number of respondents	%
It is currently in vogue	45	8.75
None	2	0.39
Ranked amongst the best	208	40.47
Demand in the labour market	78	15.18

Facilitates learning	119	23.15
Easy to use	62	12.06
Total	514	100.00

The results in Table 6 show that, out of a total of 514 teachers representing 100 per cent, the majority – 208 students, or 40.47 per cent – chose a particular language because it is ranked amongst the best languages. This was followed by 119 students (23.15 per cent) who chose a language because it makes learning easier; then 15.18 per cent (78 students) who would have liked to learn a language because it is in demand in the world of work; followed by 12.06 per cent (62 students) who would have liked to learn a language because it is easy to use; followed by 8.75 per cent of students who would have liked to learn a language because it is a current topic. Finally, 0.39 per cent, or 2 students, had no idea what would have motivated them to learn a particular language.

The chi-squared test yielded a value of 295.91 ($p = 0.000$) with 5 degrees of freedom; this leads us to reject the null hypothesis.

2. Teaching methodologies

2.1. Teaching methods used

At this stage of our survey, we sought to identify the teaching methods used by the lecturer in the programming course; the following question was asked: ‘What teaching methods do you use to teach programming in the first year of the Computer Science degree?’ The responses provided by our respondents are presented in the table below:

Table 7: Teaching methods used by the teacher.

Teaching methods	Number of students	%
Demonstration	5	41.67
Expository-interrogative	1	8.33
Interactive	2	16.67
Interactive, demonstrative	1	8.33

Interrogative	3	25.00
Total	12	100.00

Analysing Table 7, we can see that a majority of teachers (5 out of 12 teachers, or 41.67 per cent) stated that they use the demonstrative method, followed by 3 out of 12 teachers, or 25 per cent, who stated that they use the interrogative method, followed by 2 teachers, or 16.67 per cent, who use the interactive method, and finally 8.33 per cent (i.e. 1 teacher) who stated that they used a combination of the expository-interrogative method or the interactive and demonstrative methods.

The chi-squared test yielded a value of 4.667 ($p = 0.323$) with 4 degrees of freedom; this leads us to conclude that the difference is not significant.

2.2. Assessment practices

In order to ascertain the number of assignments submitted by a student, the following question was asked: ‘With regard to assessment, how many assignments did you submit for the practicals?’

Table 8: Practical Work (PW)/Student.

Number of Practical Sessions (PS)	Student numbers	%
One practical session	7	1.36
Two practical sessions	200	38.91
Three practicals	101	19.65
Four practical sessions	7	1.36
Five practical sessions	194	37.74
More than five practical assignments	5	0.97
Total	514	100.00

An analysis of Table 8 shows that, out of a total of 514 students representing 100 per cent, a majority – 200 out of 514, or 38.91 per cent – submitted more than two practical assignments in a programming language course, followed by 194 students, or 37.74 per cent, who confirmed that they had submitted five practical assignments, then 101 students (19.65 per cent) confirmed that they had submitted 3 practical assignments, followed by 7 students (1.36 per cent) who confirmed that they had submitted 1 practical assignment, and finally 5 students (0.97 per cent) who confirmed that they had submitted more than 5 practical assignments.

The chi-squared test yielded a value of 512.77 ($p = 0.000$) with 5 degrees of freedom; this leads us to reject the null hypothesis.

3. Skills acquired

Following the same line of reasoning, we sought to specify the skills expected by a lecturer at the end of a programming language course. To this end, the following question was posed: ‘What skills are expected at the end of the first-year programming language course?’. The responses provided by our respondents are presented in the table below:

Table 9: Skills expected of the teacher at the end of the course.

Expected skills	Number of students	%
Create a programme with a graphical user interface; use control structures	2	16.67
Be able to design applications	1	8.33
Be proficient in the integrated development environment	2	16.67
Producing students capable of creating high-quality software	1	8.33
Solving minor problems in procedural programming	5	41.67
Translating algorithms into programmes	1	8.33
Total	12	100.00

An examination of Table 9 shows that, out of a total of 12 lecturers representing 100 per cent, a majority of 5 lecturers (41.67 per cent) stated that they expected students to possess the following skill: ‘solving simple problems in procedural programming’, followed by 16.67 per cent of lecturers – that is, 2 lecturers – who jointly stated that they expected students to possess the following skills: (a) creating a programme with a graphical user interface and using control structures, and (b) mastering the integrated development environment; finally, 16.67 per cent (i.e. 1 teacher) stated, in conjunction with other teachers, that they expected students to possess the following skills: (a) being able to design applications, (b) producing students capable of creating good software, and (c) translating algorithms into programmes.

The chi-squared test yielded a value of 6 ($p = 0.306$) with 5 degrees of freedom; this leads us to conclude that the difference is not significant.

IV. Discussion

The analysis reveals a **discrepancy between the programming languages taught and current global standards**. Languages such as Python, ranked at the top by the PYPL and TIOBE indices, are very poorly represented in Lubumbashi. The choice of languages often appears to be ‘inherited’, with the same lecture notes being passed down year after year.

From a pedagogical perspective, the absence of active teaching methods such as the **flipped classroom** or **instructional design** limits student engagement. The widespread reliance on simply ‘repeating the explanations’ (50 per cent) when a student does not understand, rather than diversifying teaching materials, hinders the achievement of educational objectives. Furthermore, the lack of specific training for teachers in **computer science pedagogy** (75 per cent have never undertaken any) explains the persistence of these problems.

V. Conclusion

The teaching of programming at first-year undergraduate level in Lubumbashi rests on a fragile foundation, characterised by an obsolete national curriculum and traditional teaching methods. To remedy this situation, it is imperative that the government updates the curriculum by incorporating modern programming languages and that institutions train teachers in new approaches to computer science pedagogy. The adoption of methods that encourage self-directed learning is essential to make students competitive in the global market.

References

1. **Hoganson, K. E.** (2007). *Concepts in Computing*. Jones & Bartlett Learning.
2. **Hartmann, W., et al.** (2006). *Teaching Computer Science*. Springer-Verlag.
3. **Baron, G.-L., & Bruillard, É.** (2001). *A Didactics of Computer Science?* Revue Française de Pédagogie.
4. **Nsenge Mpia, H.** (2020). *Grievances regarding the failure of computer science teaching in universities in the DRC*. Etincelles.
5. **Vandeput, E., & Henry, J.** (2018). *Learning to programme: how do teachers justify their choice of a teaching tool?* Didapro7.
6. **Swinnen, G.** (2010). *Learning to programme with Python 3*. Eyrolles.