International Journal of Science and Engineering Applications
Volume 15-Issue 02, 27 – 31, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1502.1003

An Experimental Comparison of Python and Java for Beginner Programming Tasks

Zeeshan Ahmed
Student, Department of CSE,
ITER, Siksha ‘O’ Anusandhan University,
Bhubaneswar, Odisha, India

Shivangini Bihari
Assistant Professor,
Department of CSE & IT,
Jharkhand Rai University,
Ranchi, Jharkhand, India

,

Abstract: Learning to program poses important challenges for beginners, and the option of a preliminary programming language can powerfully control learning outcomes. Python and Java are broadly used in first-year computer science courses, yet they vary in syntax, arrangement, and academic approach. This study present an experimental assessment of Python and Java by beginner-level programming tasks usually taught at the undergraduate stage. Identical basic problems, counting arithmetic operations, conditional statement, loops, methods, and basic array processing, were implemented in both languages. The evaluation focus on learning-oriented metrics like lines of code, readability, programming effort, and the frequency of beginner-level errors, while excluding performance-related factors. The results specify that Python enable more concise and readable code, reducing initial learning complexity, whereas Java promotes structured and disciplined programming practice despite greater code wordiness. These findings offer realistic guidance for educators and curriculum designers in selecting suitable programming languages for preliminary programming courses.

Keywords: Python, Java, Beginner Programming, Programming Education

1. INTRODUCTION

Programming has become a primary skill in the contemporary digital era, mainly for students pursuing computer science and related disciplines. Preliminary programming courses often represent students’ first proper experience to computational thinking and problem-solving. For beginners, learning to program needs understanding new concepts like variables, control structures, and logical analysis, which can be demanding during the early stages. Consequently, the option of a suitable programming language plays a vital role in shaping a novice learner’s understanding and overall learning skill [4].

Python and Java are among the most broadly adopted programming languages in academic institutions. Python is recognized for its simple syntax and high readability, enabling beginners to focus mainly on problem-solving rather than syntactic difficulty. In contrast, Java adopts a added structured and formal approach, needing explicit data type declaration and a well-defined program structure. Although this might add to the initial learning attempt, it also encourages disciplined programming practices that are helpful for large-scale software development. Several existing studies compare Python and Java using performance-oriented metrics like execution speed, memory usage, and advanced language features; however, these factors are frequently less relevant for novice programmers. First-year undergraduate students are normally more worried with mastering fundamental concepts, counting basic input–output operations, conditional statements, loops, methods, and simple data structures. Therefore, comparisons based on beginner-level tasks give more pedagogically meaningful insights [2,3].

This paper presents an experimental comparison of Python and Java using basic programming problems commonly practiced in first-year undergraduate programming courses. The study evaluates learning-oriented parameters like lines of code, readability, programming effort, and the beginner errors. By emphasizing practical learning aspects rather than performance metrics. The findings aim to assist educators and novice programmers in making informed choices about programming languages for preliminary courses [8].

2. OBJECTIVE

The main objectives of this study are as follows:

-To experimentally compare Python and Java using beginner-level programming tasks commonly taught in first-year undergraduate courses, focusing on learning-oriented aspects.

-To analyze differences between Python and Java in terms of code length, syntactic simplicity, readability, programming effort, and common beginner-level errors.

-To provide pedagogically relevant insights that help educators and novice programmers in taking informed decisions as regards to the selection of programming languages for initial programming courses.

3. LITERATURE REVIEW

Several studies have explored comparisons between programming languages to evaluate their suitability for educational and software development contexts. Among these, Python and Java have paying attention considerable attention due to their extensive adoption in both academia and industry[9]. Early research has mainly focused on performance-oriented parameters, like execution time, memory utilization, and computational competence, to assess the strengths and boundaries of these languages[4].

A number of studies recognize Python as a beginner-friendly programming language because of its easy syntax, dynamic typing, and reduced code length. This uniqueness enables novice programmers to focus more on problem-solving rather than syntactic details. Research in programming education reports that students introduced to Python often reveal faster initial growth and greater confidence during early programming courses. Python's high readability and close similarity to natural language are frequently cited as key factors contributing to its effectiveness in introductory programming curricula [1].

On the contrary, several researchers stress the pedagogical advantages of Java in teaching structured and object-oriented programming concepts. Java's strict syntax rules, explicit data type declarations, and compile-time error checking are measured beneficial for developing strong initial programming skills. Although beginners may initially recognize Java as more challenging, studies recommend that it encourages disciplined coding practices and a clearer understanding of program structure and organization [2].

Despite these findings, a noteworthy constraint of existing literature is that many comparative studies between Python and Java rely on advanced algorithms, performance benchmarks, or higher-level programming features. Such approaches may not precisely reflect the learning experience of first-year undergraduate students, who usually engage with basic tasks concerning input/output operations, conditional statements, loops, methods, and simple data structures[6,10]. Few numbers of studies provide experimental comparisons focused specifically on beginner-level programming tasks commonly practiced in introductory laboratories [3, 4].

This gap in the literature motivates the present study. By emphasizing basic programming exercises and learning-oriented parameters, this research seeks to offer a more practical and learner-centric comparison of Python and Java, thereby contributing meaningful insights to introductory programming education.

4. OVERVIEW

Programming languages provide a formal means for expressing computational instructions in a structured and logical manner. At the introductory level, an effective programming language should emphasize clarity, ease of comprehension, and the progressive development of problem-solving skills. Novice learners typically engage with fundamental programming constructs, including variables, input and output operations, conditional statements, loops, methods, and basic data structures [7, 8].

Python and Java are widely adopted programming languages for teaching these foundational concepts in introductory programming courses. While both languages are capable of addressing similar problem domains, they differ substantially in terms of syntax design, execution model, and pedagogical approach. Understanding these differences is essential for assessing their suitability for beginner programmers. Accordingly, this section presents a brief overview of Python and Java to establish the conceptual foundation for the experimental comparison conducted in this study [9].

4.1 Overview of Java

Java is a high-level, object-oriented programming language widely used in academic, enterprise, and industrial applications. It follows a strict syntactic structure and requires explicit declarations of data types, classes, and methods. Java programs are compiled into platform-independent bytecode and executed on the Java Virtual Machine (JVM), enabling portability across different computing environments [9].

From an educational perspective, Java emphasizes structured programming and core object-oriented principles such as encapsulation and modularity. Novice programmers using Java must understand formal program constructs, including class definitions, method signatures, and variable declarations. Although this requirement may increase the initial learning effort, it promotes disciplined coding practices and a strong understanding of program organization [6,7].

Additionally, Java performs extensive compile-time error checking, allowing many programming errors to be detected prior to execution. This feature is particularly beneficial for beginners, as it encourages careful program design and early error identification. Owing to these characteristics, Java is frequently adopted in introductory programming courses aimed at establishing a strong and systematic programming foundation [8].

4.2 Overview of Python

Python is a high-level, interpreted programming language widely recognized for its syntactic simplicity and high readability. Its code structure closely resembles natural language, which facilitates comprehension of program logic among novice programmers. Python employs dynamic typing, thereby removing the requirement for explicit variable type declarations [4].

In the context of introductory programming education, Python enables beginners to focus primarily on problem-solving rather than syntactic intricacies. Fundamental programming tasks, including input handling, arithmetic computations, and control flow constructs, can be implemented using fewer lines of code compared to many other programming languages. This conciseness contributes to faster initial learning and increased learner confidence[5].

Python programs are executed through an interpreter, providing immediate feedback during execution. Although errors are detected at runtime, this interactive execution model supports experimentation and exploratory learning. Owing to its ease of use, flexibility, and wide applicability, Python is extensively adopted as a first programming language in academic institutions [7, 8].

5. METHODOLOGY

This study adopts an experimental methodology to compare Python and Java from a beginner-oriented learning perspective. Identical beginner-level programming tasks were implemented in both languages and evaluated using learning-focused criteria. The methodological design emphasizes fairness, simplicity, and relevance to introductory programming education.

5.1 Experimental Approach

An experimental approach was employed in which the same logical problems were independently implemented using Python and Java. The problem-solving logic was kept strictly consistent across both implementations to ensure that any observed differences could be attributed solely to language-specific characteristics rather than algorithmic variations.

Only fundamental programming constructs were used in both languages. Advanced language features, external libraries, and optimization techniques were intentionally excluded to maintain a beginner-focused and concept-oriented comparison.

5.2 Programming Tasks

The study considers a set of basic programming tasks that represent common exercises encountered in introductory programming courses. These tasks are designed to assess fundamental programming concepts', including arithmetic operation, conditional statements, looping constructs, methods, and basic array processing.

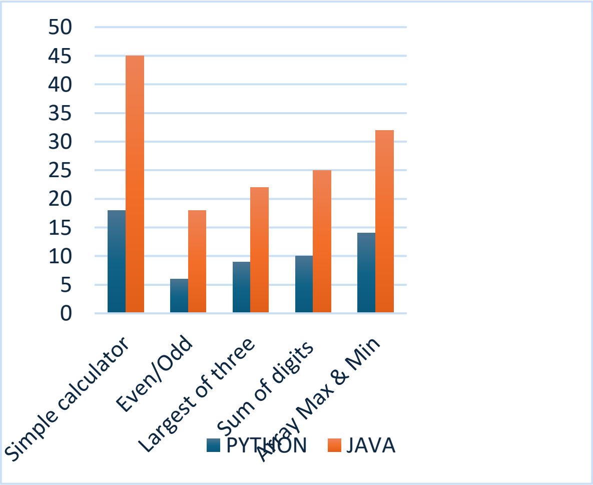
The selected programming tasks are as follows:

- Simple calculator using methods
- Even or odd number checking
- Determining the largest of three numbers
- Calculating the sum of digits of a number
- Identifying the maximum and minimum elements in an array

These tasks are intentionally simple and reflective of early-stage programming exercises. They were chosen to evaluate how beginner programmers interact with Python and Java when applying core programming constructs commonly introduced during the initial phases of learning.

5.3 Implementation Detail

Fig.1. Implementation Detail



All programs were implemented using straightforward syntax and constructs appropriate for beginner programmers. In both languages, clarity and correctness were prioritized over performance optimization.

JavaImplementation

Java programs were developed using basic class and method structures. Standard input and output mechanisms were employed, and the programs were compiled and executed in a standard Java runtime environment.

Python Implementation

Python programs were written using Python 3 syntax with simple input handling, control flow, and basic data manipulation constructs. The programs were executed using a standard Python interpreter.

This consistent implementation approach ensured a fair comparison between Python and Java, focusing exclusively on learning-related characteristics.

5.4 Evaluation Criteria

The comparison between Python and Java was conducted using learning-oriented evaluation criteria that are particularly relevant to beginner programmers. The selected criteria are defined as follows:

- **Lines of Code (LOC):** The total number of lines required to implement each programming task.
- **Syntax Simplicity:** The ease with which beginners can write and comprehend the syntax of the programming language.
- **Readability:** The degree to which program logic is easily understandable by novice programmers.
- **Programming Effort:** The amount of structural and syntactic effort required to develop a correct and functional solution.
- **Potential Beginner Errors:** The likelihood of encountering common syntactic or logical errors during program development.

Performance-related metrics, such as execution time and memory consumption, were intentionally excluded, as they are not central to evaluating beginner-level learning experiences.

5.5 Data Analysis

Observations for each programming task were systematically recorded and organized into comparative tables. The collected data were analyzed to identify patterns related to code length, readability, and overall learning effort. The analysis focuses on highlighting practical and pedagogically relevant differences between Python and Java from the perspective of novice programmers.

6. Results and Discussion

This section presents the results obtained from the experimental comparison of Python and Java using beginner-level programming tasks. The discussion focuses on learning-oriented aspects such as code length, syntax simplicity, readability, and potential beginner difficulties.

6.1 Comparison Based on Lines of Code

Table 1 : Comparison Based on Lines of Code

Programming Task	Python(LOC)	Java (LOC)
Simple calculator	18	45
Even/odd numbers	6	28
Largest of three numbers	9	22
Sum of digits	10	25
Array Max & Min	14	32

One of the most prominent differences observed in the experiments is the number of lines of code required to implement identical programming tasks. Across all selected tasks, Python programs consistently required fewer lines of code than Java programs. This reduction is primarily attributable to Python’s concise syntax, the absence of explicit type declarations, and simplified input and output handling.

In contrast, Java programs require additional structural components, including class definitions, method declarations, and explicit data type specifications, which increase overall code length. Consequently, beginners using Python can complete programming tasks with less writing effort, whereas Java necessitates a more detailed and structured program design. **Table 1** summarizes the differences in lines of code required for Python and Java implementations.

Table 2: Syntax Simplicity and Programming Effort

Parameters	Python	Java
Variable declaration	Not required	Required
Code verbosity	Low	High
Input handling	Simple	More structured
Program structure	Minimal	Class-based
Overall effort	Low	Moderate

6.2 Syntax Simplicity and Programming Effort

Python exhibits greater syntactic simplicity for beginner programmers, as fundamental tasks such as arithmetic operations, conditional statements, and looping constructs can be expressed using minimal keywords and symbols. This simplicity reduces the initial cognitive load and allows learners to focus more on problem-solving rather than syntax.

Table 3: Common Beginner Errors in Python and Java

Aspect	Python	Java
Syntax errors	Indentation issues	Missing semicolons
Type-related	Runtime type errors	Compile-time type errors
Error detection	Mostly runtime	Mostly compile-time
Debugging ease	Moderate	Easier for beginners

In contrast, Java, while more verbose, enforces a well-defined and structured programming paradigm. Beginners must strictly adhere to syntax rules involving semicolons, brackets, and explicit variable declarations. Although this increases the initial learning effort, it fosters an understanding of program structure and disciplined coding practices. Consequently, Python reduces syntactic complexity, whereas Java emphasizes formalism and structural rigor in programming.

6.3 Readability and Ease of Understanding

Readability plays a crucial role in the learning experience of novice programmers. Python programs closely resemble natural language and pseudo code, making them easier for beginners to read and comprehend. The use of indentation to define code blocks provides a clear visual representation of program flow, enabling learners to grasp program logic more intuitively.

Java programs, in contrast, may initially appear complex due to their verbose syntax and strict structural requirements. However, once learners become familiar with the syntax, Java’s explicit organization—through clearly defined classes, methods, and block delimiters—offers a structured and unambiguous representation of program execution.

Overall, Python provides higher immediate readability and ease of understanding for beginners, while Java enhances clarity through explicit structural definitions as learners gain proficiency.

6.4 Potential Beginner Errors

During the implementation of the selected programming tasks, distinct categories of beginner-level errors were observed in both Python and Java. These errors reflect language-specific characteristics that influence the learning and debugging experiences of novice programmers.

In Java, common beginner errors include missing semicolons, incorrect data type declarations, and improper placement of braces. Java’s strong compile-time error detection identifies many of these issues before program execution, facilitating early error recognition and encouraging disciplined coding practices among beginners.

In Python, frequently observed errors involve incorrect indentation and type-related issues during input handling and variable usage. As Python is an interpreted language, many errors are detected at runtime, which directly affects the debugging process. This execution model promotes iterative learning, allowing beginners to experiment, observe immediate feedback, and refine their understanding through trial and error.

Overall, Java’s compile-time checking supports early and systematic error detection, whereas Python’s runtime error reporting fosters exploratory learning. These differences significantly influence how beginners identify, interpret, and learn from programming mistakes.

6.5 Overall Discussion

The experimental findings indicate that Python is more effective in reducing initial learning difficulty due to its concise syntax, minimal structural requirements, and ease of use. These characteristics facilitate faster comprehension of fundamental programming concepts and promote early confidence among beginner programmers.

In contrast, Java demands greater initial effort because of its explicit syntax, strict type declarations, and structured program organization. While this increases the learning overhead at the introductory stage, it encourages disciplined coding practices and helps learners develop a strong understanding of program structure and control flow.

Overall, Python is well suited for rapid skill acquisition and introductory learning environments, whereas Java is advantageous for establishing a robust and structured programming foundation that supports long-term programming competence.

7. CONCLUSION

This paper presented an experimental comparison of Python and Java using beginner-level programming tasks commonly introduced in introductory programming courses. The analysis focused on learning-oriented parameters, including lines of code, syntax simplicity, readability, programming effort, and potential beginner errors, while deliberately excluding performance-related metrics to maintain relevance for novice learners.

The results demonstrate that Python generally requires fewer lines of code and offers simpler syntax, enabling easier comprehension and implementation of basic programming constructs. This simplicity reduces initial learning difficulty and allows beginners to focus primarily on problem-solving rather than syntactic complexity. Conversely, Java, despite being more verbose, enforces structured programming practices through explicit declarations and clearly defined program organization, thereby promoting disciplined and systematic coding habits.

Based on these findings, Python is well suited for learners seeking rapid conceptual understanding and early confidence building, while Java is more appropriate for developing strong foundational and structured programming skills. The study concludes that both languages are effective for introductory programming education, and the choice of language should be guided by instructional objectives and curriculum design rather than perceived overall superiority.

8. FUTURE SCOPE

The present study is limited to basic programming tasks and beginner-level concepts. Future research may extend this work in several directions. First, additional programming languages, such as C++ or JavaScript, can be included to enable broader comparative analysis. Second, empirical investigations based on real student performance, learning outcomes, and classroom data would provide stronger educational evidence.

Further extensions may involve analyzing debugging effort and the time required to identify and correct errors, as well as expanding the comparison to include object-oriented programming concepts. Additionally, future studies could evaluate the long-term impact of programming language selection on programming proficiency and skill retention. Such extensions would provide deeper and more comprehensive insights into the role of programming languages in computer science education.

9. REFERENCES

[1] G. van Rossum and F. L. Drake, *The Python Language Reference Manual*. Bristol, U.K.: Network Theory Ltd., 2011.

[2] H. Schildt, *Java: The Complete Reference*, 11th ed. New York, NY, USA: McGraw-Hill Education, 2018.

[3] T. R. Henry and R. A. Lahey, "A comparison of programming languages for introductory computer science courses," *Journal of Computing Sciences in Colleges*, vol. 26, no. 6, pp. 154–160, 2011.

[4] A. Gupta and R. Verma, "Comparative analysis of Python and Java for beginners," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 5, pp. 2395–2402, 2020.

[5] M. Irfan Alam and S. N. Singh, "Designing and Implementing cloud security using multi-layer DNA cryptography in python," in *Trends in Wireless Communication and Information Security*, M. Chakraborty, R. K. Jha, V. E. Balas, S. N. Sur, and D. Kandar, Eds., Lecture Notes in Electrical Engineering, vol. 740. Singapore: Springer, 2021, doi: 10.1007/978-981- 33-6393-9_38.

[6] S. Bihari and M. I. Alam, "Leveraging recommender systems for course selection in higher education: A pathway to informed decision-making," in *Proceedings of the Recent Advances in Artificial Intelligence for Sustainable Development (RAISD 2025)*, *Advances in Intelligent Systems Research*, Atlantis Press (Springer Nature), 2025, doi: 10.2991/978-94-6463- 787-8_27.

[7] A. Azenkouk, "Python vs Java: Which is better for beginners?" *International Journal of Research and Innovation in Social Science (IJRISS)*, vol. 9, no. 8, pp. 2392–2397, 2025.

[8] M. Naveed, "Pedagogical suitability: A software metrics-based analysis of Java and Python," *International Journal of Innovations in Science & Technology*, 2025.

[9] J. Karnando, M. Muskhir, and A. Luthfi, "Exploring Python programming: A project-based learning-centric experience," *Journal of Education Technology*, vol. 8, no. 2, pp. 306–314, 2024.

[10] I. Alam, "Enhancing cloud security using multi-level DNA cryptography," *Splint International Journal of Professionals*, vol. 7, no. 1, pp. 75–82, 2020.