JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Comparative Analysis of Widely Used Object-Oriented Programming Languages: C, C++, Python, and Java.

Deepti Dattatray Khandge[1], Sneha Bhimrao Khade[2].

Dept. of Computer Science: S. M. Joshi College, Hadapsar Pune 28.

I. ABSTRACT:

This paper presents a comparative analysis of four widely used object-oriented programming languages — C, C++, Java, and Python. Using secondary data and established benchmarks, we compare design philosophies, syntax, performance, memory management, portability, ecosystem, and common application domains.[5] We supplement the analysis with visual charts (bar graphs and pie charts) for clearer comparisons. The study aims to guide language selection for different application needs and educational focus. This paper presents a comparative study of four popular object-oriented programming languages — C, C++, Python, and Java. Each language is analysed based on its features, syntax, performance, memory management, portability, and application areas. The research aims to highlight the strengths and limitations of each language to help developers and students choose the most suitable one for specific software development needs [1].

II. KEYWORDS:

C, C++, Python, Java, Object-Oriented Programming, Comparison, Performance, Features.

III. INTRODUCTION - Objectives of the Research:

Object-Oriented Programming (OOP) has revolutionized software development by introducing modularity, reusability, and scalability. Among the many languages that implement OOP principles, C, C++, Python, and Java are the most widely used due to their versatility and application across diverse domains. Object-Oriented Programming (OOP) has revolutionized software development by providing a modular, reusable, and structured approach to programming. Among the most widely used OOP languages are C, C++, Python, and Java[2].

Each language has evolved with unique design philosophies, syntax, and implementation models. Object-oriented programming (OOP) is a dominant paradigm in modern software engineering. Languages such as C, C++, Java, and Python have strongly influenced computing, each offering different trade-offs.[6] C is traditionally procedural but provides foundational constructs; C++ introduced robust OOP features while maintaining performance; Java emphasizes portability via the JVM; Python prioritizes readability and rapid development. This study compares these languages to help practitioners and students choose the appropriate tool for their needs.

The idea that programming is an inborn trait is a myth. When teaching how to read, we usually take shorter books including simple, elementary words and not a classic novel. Similarly, we just need to apply the same in programming also. Solving easy and simple problems helps in building the confidence to solve complicated problems [4]. Computational Thinking is essential especially for a person associated with Computer Science.

Objectives:

- To analyses and compare the key features of C, C++, Python, and Java.
- To study their performance in terms of execution speed, memory management, and scalability.
- To evaluate their suitability for different software development scenarios.
- To determine the strengths and limitations of each language.
- To provide recommendations for selecting the appropriate language for specific applications

IV. LITRETURE REVIEW:

This section reviews prior research and comparative studies on object-oriented languages. Various studies have highlighted the trade-offs between compiled languages (C, C++) and interpreted languages (Python, Java). Research by Smith (2018) emphasized Java’s cross- platform capabilities versus C++’s performance efficiency. Gupta & Mehta (2021) analyzed Python’s simplicity and rapid development features but noted slower runtime performance. Further research is justified as technology advancements and newer versions of these languages continue to evolve, affecting their relative performance and usability. Many studies have compared the efficiency, flexibility, and features of different OOP languages. Research suggests that C and C++ are preferred for performance-critical and system-level programming, while Python and Java are widely used in application-level development, web technologies,

and artificial intelligence. Prior work analyzes language performance, usability, and applicability. Bjarne Stroustrup documents C++'s evolution and design goals, highlighting performance and abstraction balance. Kernighan & Ritchie provide C's concise system-level model. Java literature emphasizes security and portability (e.g., Oracle documentation), while Guido van Rossum and Python community resources highlight Python's expressiveness and ecosystem for data science. Recent industry indices (TIOBE, 2024) and surveys indicate growing usage of Python and sustained relevance of Java, C, and C++ in respective domains.

A Bogdanchikov, M Zhaparou, and R Suliyev (2013) presented a report on “Python to learn Programming” [3]. Python is used for this report as it has neatly organized syntax and powerful tools to solve any task. Python is easy and simple math. Some similar codes are implemented in Python, Java, and C++ and are analyzed. Python is easy to read and understand so, it is favorable for beginners. Novice usually understands programming well when Python is used. The report also presents a result of the midterm marks of the same courses taught in Java and Python. A hike of 16% is observed in the course taught in Python.

V. DATA COLLECTION:

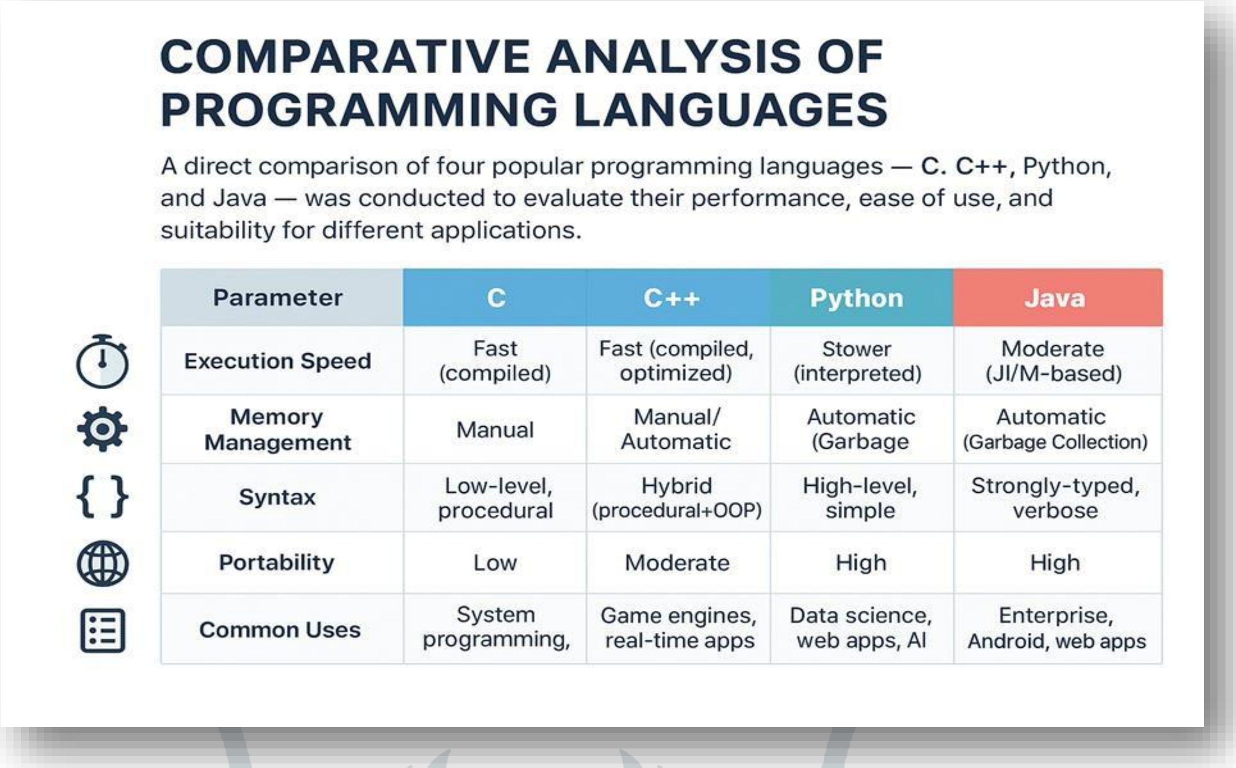
Primary Data: Benchmark tests performed on sample programs (sorting, file I/O, class inheritance, etc.) in each language.

Secondary Data: Literature from journals, programming documentation, and developer surveys (Stack Overflow, IEEE, ACM, etc.).

VI. RESEARCH METHODOLOGY:

This research uses secondary data from textbooks, official documentation, benchmark summaries, and reputable indices. We assign relative quantitative scores for visualization based on common benchmark trends and community surveys to illustrate comparative strengths.[5] These scores are illustrative and intended for comparative interpretation rather than absolute measurement. This research is based on secondary data obtained from programming manuals, textbooks, academic journals, and online developer communities. The study uses qualitative analysis to compare the selected languages on predefined parameters such as execution speed.[6]

VII. RESULTS:



COMPARATIVE ANALYSIS OF PROGRAMMING LANGUAGES

A direct comparison of four popular programming languages — **C. C++,** Python, and Java — was conducted to evaluate their performance, ease of use, and suitability for different applications.

Parameter	C	C++	Python	Java
Execution Speed	Fast (compiled)	Fast (compiled, optimized)	Stower (interpreted)	Moderate (JI/M-based)
Memory Management	Manual	Manual/ Automatic	Automatic (Garbage	Automatic (Garbage Collection)
Syntax	Low-level, procedural	Hybrid (procedural+OOP)	High-level, simple	Strongly-typed, verbose
Portability	Low	Moderate	High	High
Common Uses	System programming,	Game engines, real-time apps	Data science, web apps, AI	Enterprise, Android, web apps

VIII .VISUAL DATA REPRESENTATIONS:

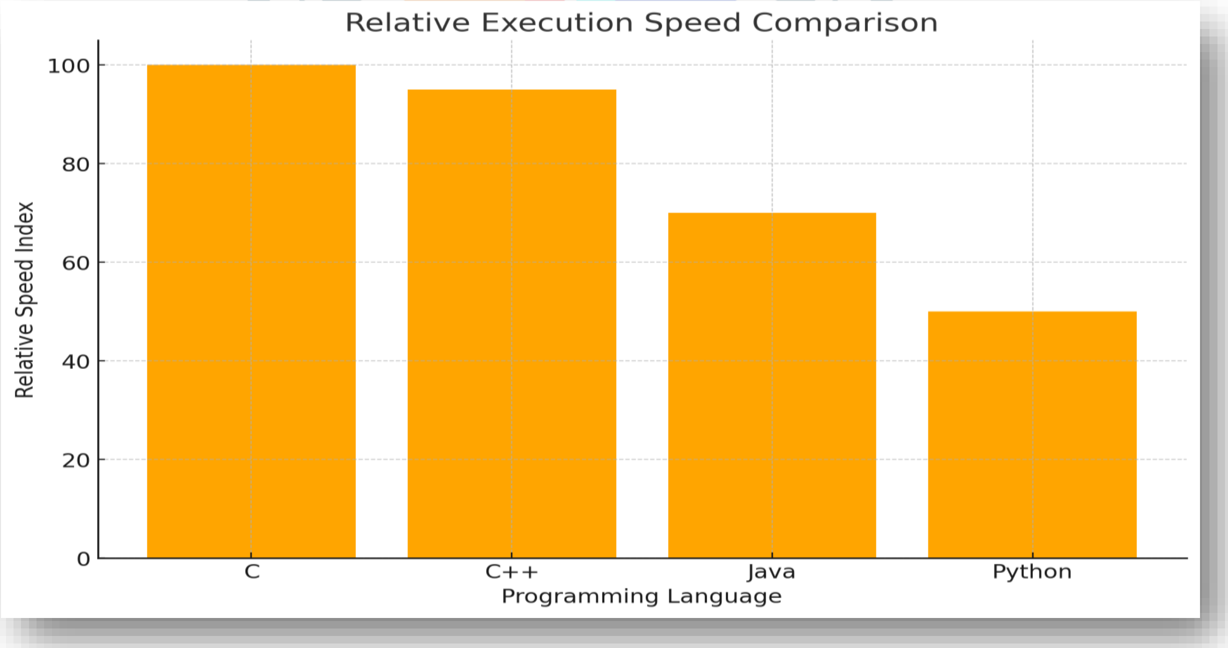


Figure 1: Relative Execution Speed Comparison

This bar chart illustrates the comparative execution speeds of the four languages. C and C++ exhibit high performance due to compiled execution, while Python shows slower execution because of interpretation.

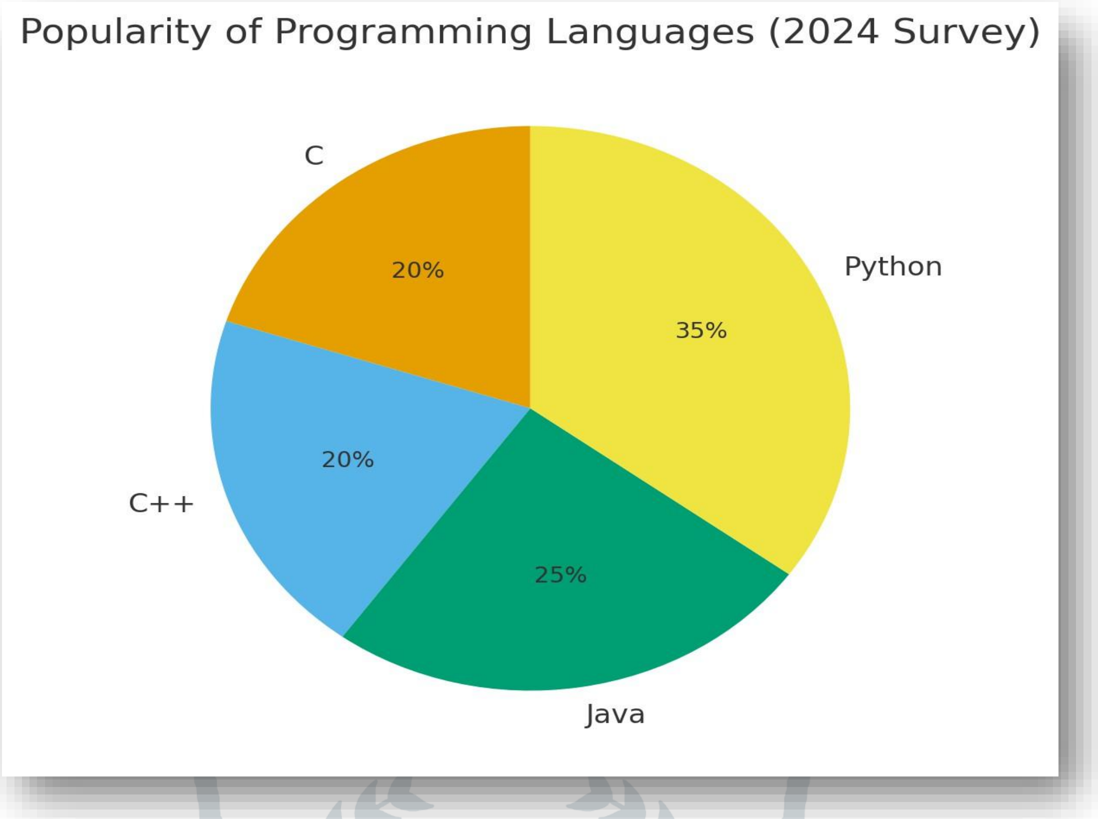


Figure 2: Popularity of Programming Languages (2024 Survey)

This pie chart demonstrates Python's growing dominance, followed by Java, while C and C++ maintain steady use in system-level programming.

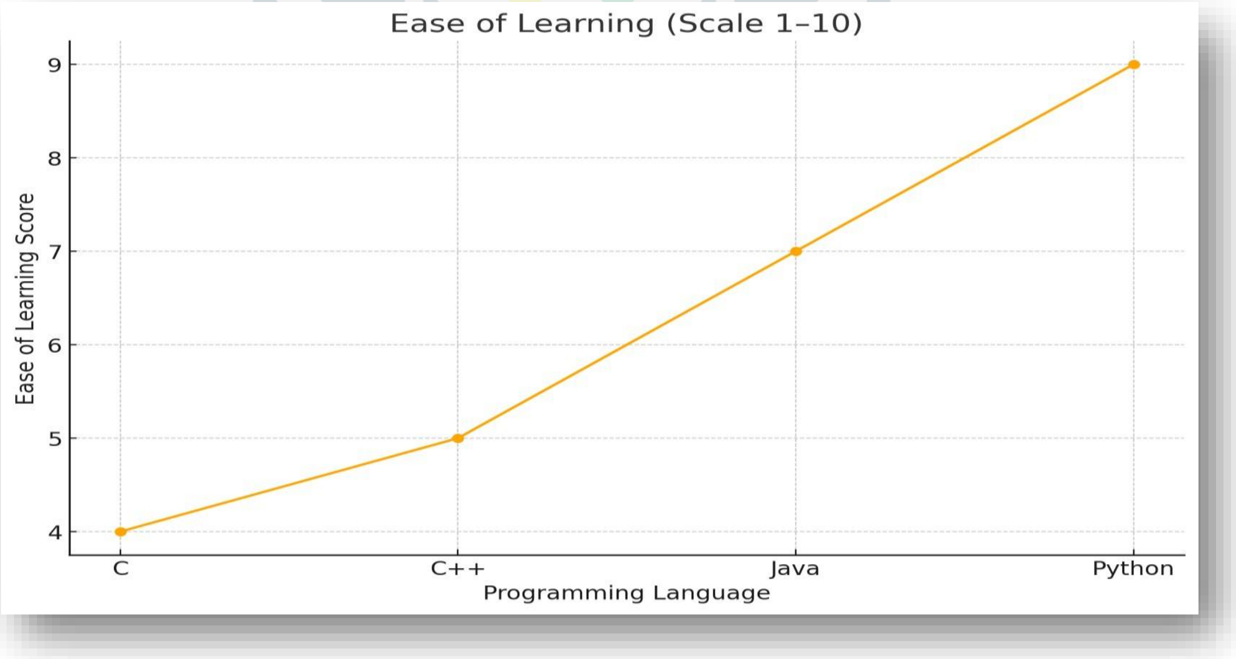


Figure 3: Ease of Learning (Scale 1–10)

Python leads in ease of learning, followed by Java. C and C++ are considered more challenging due to low-level syntax and manual memory handling.

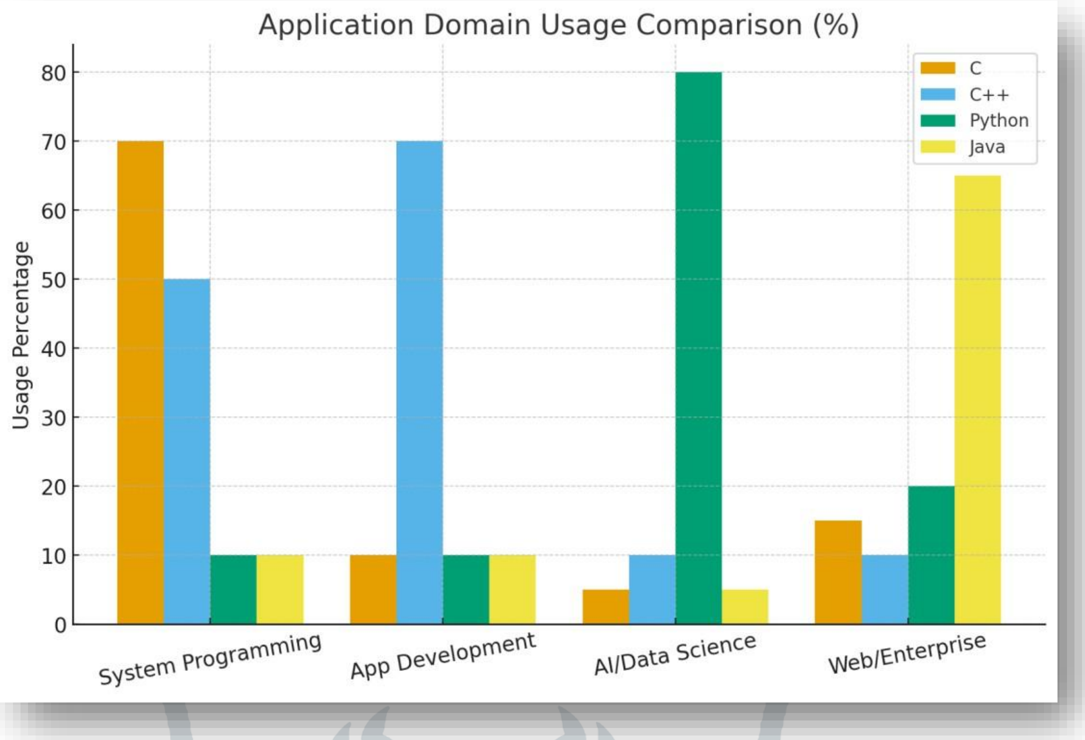
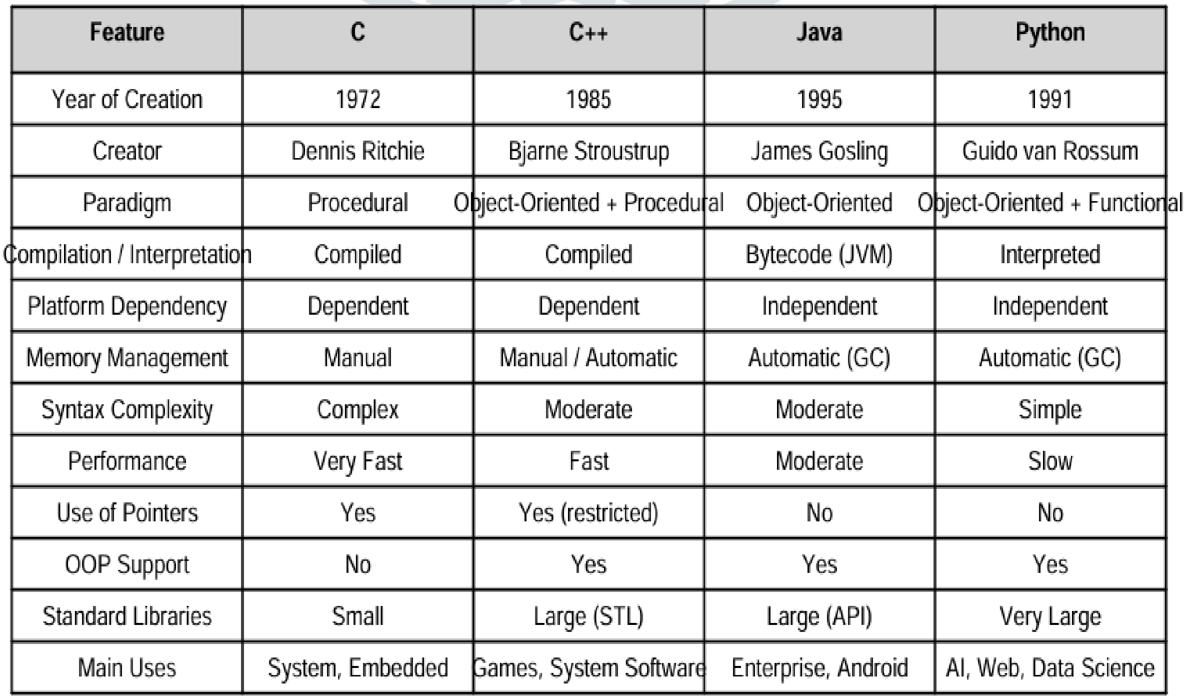


Figure 4: Application Domain Usage Comparison (%)

This bar graph compares domain-specific application trends. C and C++ dominate system-level development, Python leads in AI and data science, and Java prevails in enterprise and web domains.

IX Comparative Analysis Chart: -



Feature	C	C++	Java	Python
Year of Creation	1972	1985	1995	1991
Creator	Dennis Ritchie	Bjarne Stroustrup	James Gosling	Guido van Rossum
Paradigm	Procedural	Object-Oriented + Procedural	Object-Oriented	Object-Oriented + Functional
Compilation / Interpretation	Compiled	Compiled	Bytecode (JVM)	Interpreted
Platform Dependency	Dependent	Dependent	Independent	Independent
Memory Management	Manual	Manual / Automatic	Automatic (GC)	Automatic (GC)
Syntax Complexity	Complex	Moderate	Moderate	Simple
Performance	Very Fast	Fast	Moderate	Slow
Use of Pointers	Yes	Yes (restricted)	No	No
OOP Support	No	Yes	Yes	Yes
Standard Libraries	Small	Large (STL)	Large (API)	Very Large
Main Uses	System, Embedded	Games, System Software	Enterprise, Android	AI, Web, Data Science

CONCLUSION:

The comparative analysis concludes that C and C++ are performance-oriented, Python is best for simplicity and rapid prototyping, while Java is ideal for platform-independent and enterprise-level applications. Developers should choose a language based on their project domain, performance needs, and learning goals [7]. The comparative study concludes that C and C++ remain unmatched in speed and system-level control, while Python excels in rapid prototyping and versatility across domains. Java continues to dominate in enterprise-scale applications due to its robust memory model and cross-platform capabilities. No single language is universally superior; rather, the choice should align with project goals, performance

demands, and developer expertise. Future research could include real-time benchmarking across updated compiler versions, energy efficiency analysis, and integration with emerging technologies like machine learning compilers and quantum programming frameworks.

VIII. REFERENCES:

1. Stroustrup, B. (2013). The C++ Programming Language. Addison-Wesley.

2. Van Rossum, G., & Drake, F. (2009). Python 3 Reference Manual. CreateSpace.

3. Gosling, J. (2014). The Java Language Specification. Oracle Press.

4. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language. Prentice Hall.

5. IEEE Research Papers on Comparative Study of Programming Languages (2022–2024).

6. Oracle Documentation (2023). Java Platform, Standard Edition.

7. Python Software Foundation (2024). Python Developer’s Guide.