A Review Paper on Developer's Choice: Java or C++

Gangandeep Singh, Sukhjinder Sagga
RIMT University, Mandi Gobindgarh, Punjab

ABSTRACT: Programming languages have always been a subject of discussion or debate as in which language is best among several programming languages available in programming field. There is no exact answer to this question as all the language are best in their own terms and should be used in different projects depending upon the project requirements and by choosing which language will best suit those requirements it becomes easy to go with a particular language. In order to solve this problem of confusion, this paper has been introduced with the aim of providing main differences between Java and C++. In this review paper several concepts of Java and C++ has been discussed such as code execution in Java and C++, which language is easier to learn and understand. This paper provides a clear view of which language is best according to several needs and requirements among Java and C++. There are many opportunities in the field of programming as one can become a developer or a programmer. So, the future of programming languages will be bright as the demand of engineers has been continuously increasing due to the growth in technology.

KEYWORDS: Code, Compiler, C++, Java, Programming.

1. INTRODUCTION

A relatively new programming language, Java has attracted a lot of attention in its short-term. This is due to a variety of causes. The debut of Java in 1995 was unprecedented in terms of the amount of publicity given to a programming language at that time. Due to an upsurge in Internet usage about 1990, there was great interest in the Applets written in Java. Computer Science programmes taught C++ before Java showed out, and it quickly became the most popular language in the field. In the 1995-1996 academic year, 36 percent of authorised computer science programmes utilised Pascal as the first language taught in their curricula, while 32 percent used C++ as the first language. In the 1997-1998 academic year, Java was the first language taught in 9 percent of approved Computer Science programmes. C++ usage increased to 47 percent in the same academic year, whereas Pascal usage decreased to 6 percent[1], [2].

From 1999-2000 academic year, Java grew to 22%, C++ to 54%, while Pascal decreased to 2%. According to McCauley and Manaris' [1999] research, C++ and Java are two of the most common programming languages used in university courses. Job ads on dice.com, a prominent on-line recruitment firm, revealed that Java and C++ are highly sought-after technical abilities. Java and C++ were mentioned in 14,920 (10.1%) of the 147,875 job listings on March 14, 2002. SQL (11.7 percent) and Oracle (9.6 percent) were also cited a lot[3]. These two programming languages are widely utilised in industry because of their outstanding capabilities. Because of its efficiency, fast speed, and dynamic memory use, C++ is a frequently used programming language. Programming languages like Java are unmatched when it comes to software creation[4].

1.1 Which is better: C ++ or Java?

All that matters when it comes to deciding between Java and C++ is what features user need for your project and code. There are pros and cons to both of these languages, and one may make a decision based on this information. When it comes to system development, C++ is the way to go. Java, on the other hand, is a wonderful choice for desktop, web, and other applications[5]. Using C++, programmers may accomplish anything from system and corporate programming to game development. Programmers can't use the Java language to construct some low-level games and other applications. In other words, the choice of a programming language is completely determined by what kind of application is being created. One might develop a list of the advantages and negatives of each languages to help them decide. In this way, the user may determine what program's characteristics are ideal and choose the language accordingly[6].

1.2 Code execution in C++:

When utilising C++, the code is executed by a compiler. When the source code is compiled, it is converted to machine code by a C++ compiler. Because of this, C++ is quicker than Java, but it is not platform-independent like Java and other languages[7]. Figure 1 illustrates the execution of code into C++.

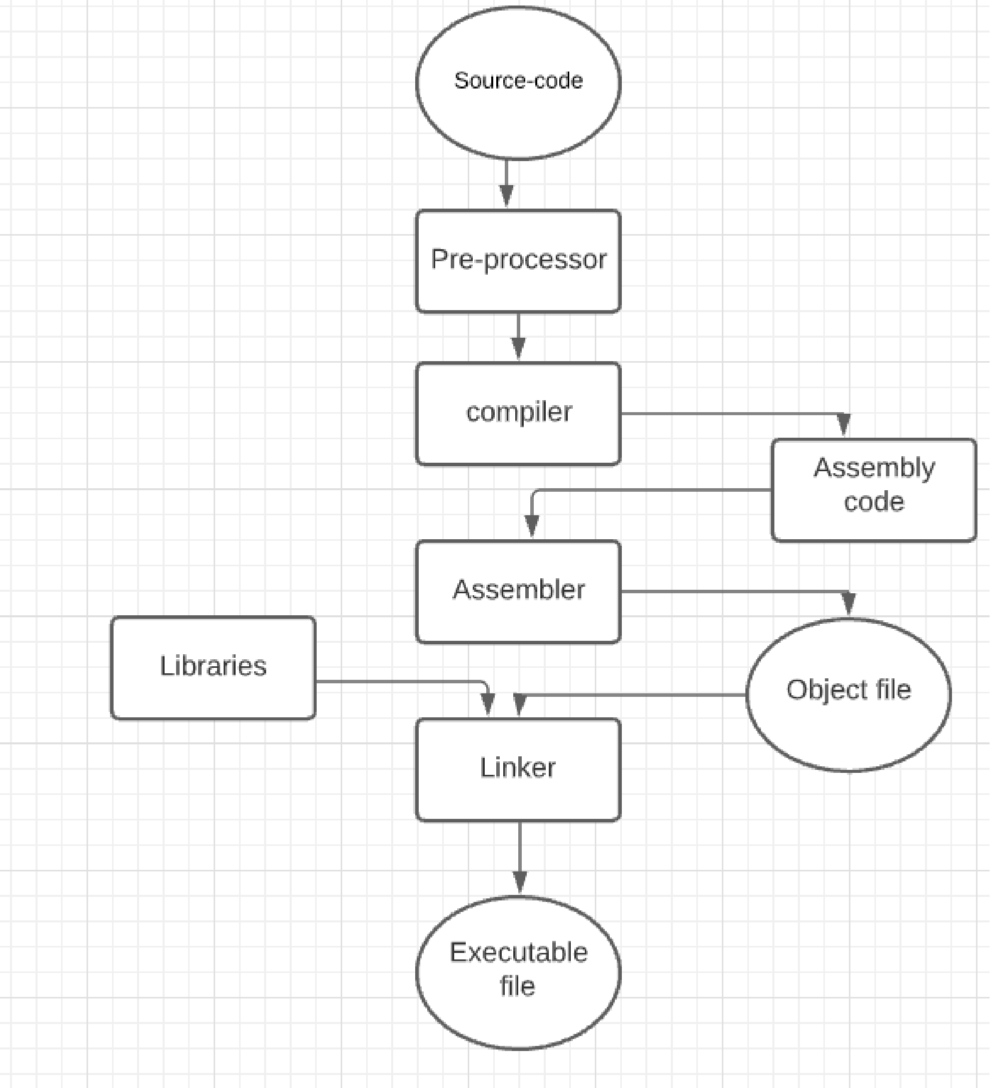


Figure 1: Representation of execution of code into C++.

1.3 Code Execution in Java:

Java source code or a.java file is transformed to byte code or a .class file during compilation. As soon as JVM (Java Virtual Machine) loads the .class file, an interpreter will transform it into machine code. JVM will run the optimised code after compiling method calls using a Just-In-Time (JIT) compiler. Because of this, java is considered as both compiled as well as an interpreted language[8]. Figure 2 illustrates execution of code into Java.

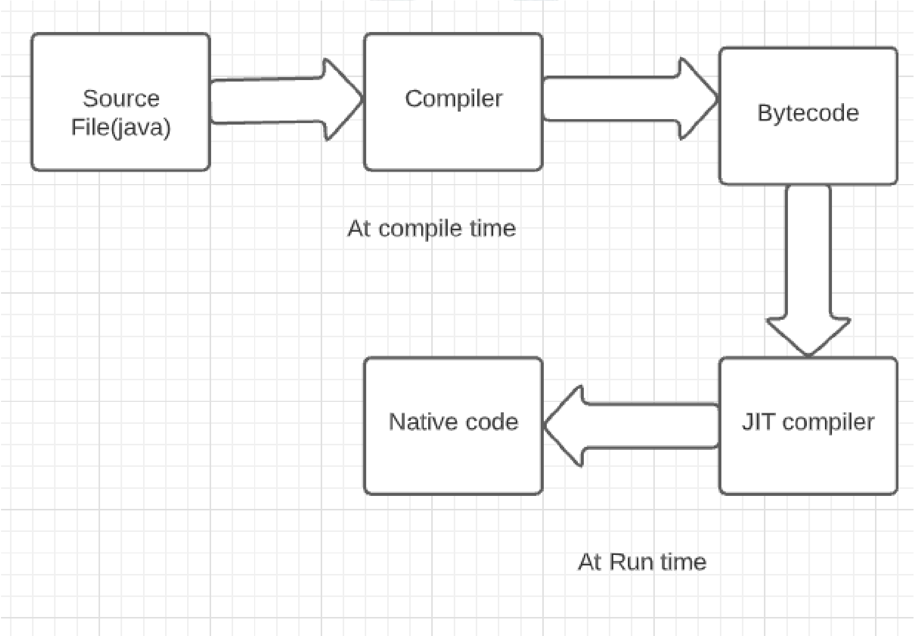


Figure 2: Representation of execution of code into Java.

1.4 Which is easier to learn and understand?

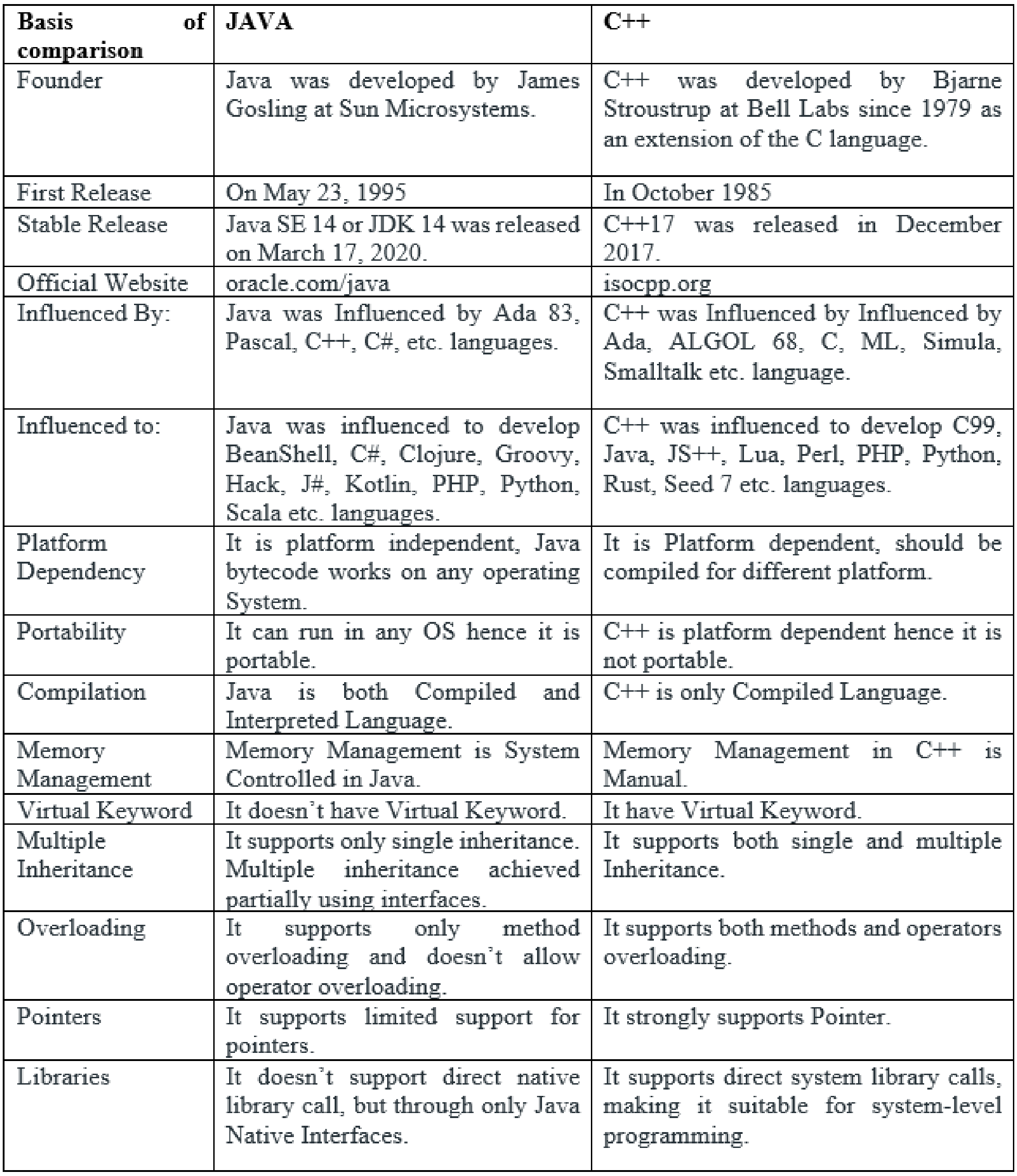
In terms of language and grammar, Java is easy to learn and utilise. In terms of low-level applications such as system programming, however, C++ takes the top spot. Java's automated garbage collection and the absence of multiple inheritance and pointers make it more powerful, while C++ is more powerful in terms of performance. In addition, the trash collection feature might make things worse in games and other applications where the state has to be preserved. So, the more powerful language may be decided based on what application you need to create and what features you are searching for[9].

1.5 Similarities between Java and C++:

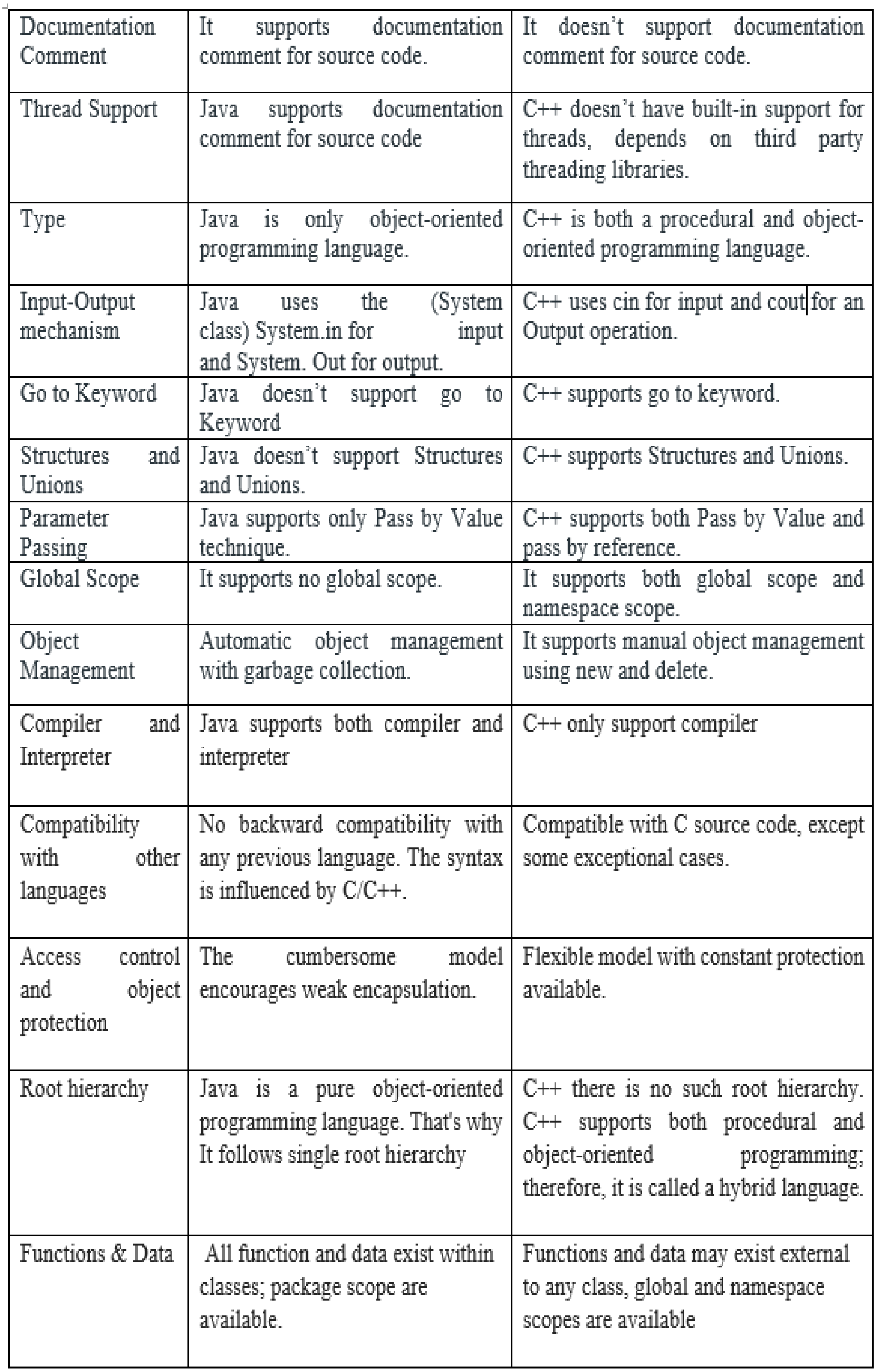
- In both C++ and Java, Object-Oriented Programming is supported.
- They have a similar syntax.
- There is no difference between their comments syntaxes: Conditional statements and loops (like if-else, switch, etc.) are identical.
- Arithmetic and relational operators are the same.
- The main function is where both C++ and Java programmes begin to execute.
- They both use the same primitive data types for their data.

Table 1 represents several dissimilarities between Java and C++ based on which these two can be differentiated.

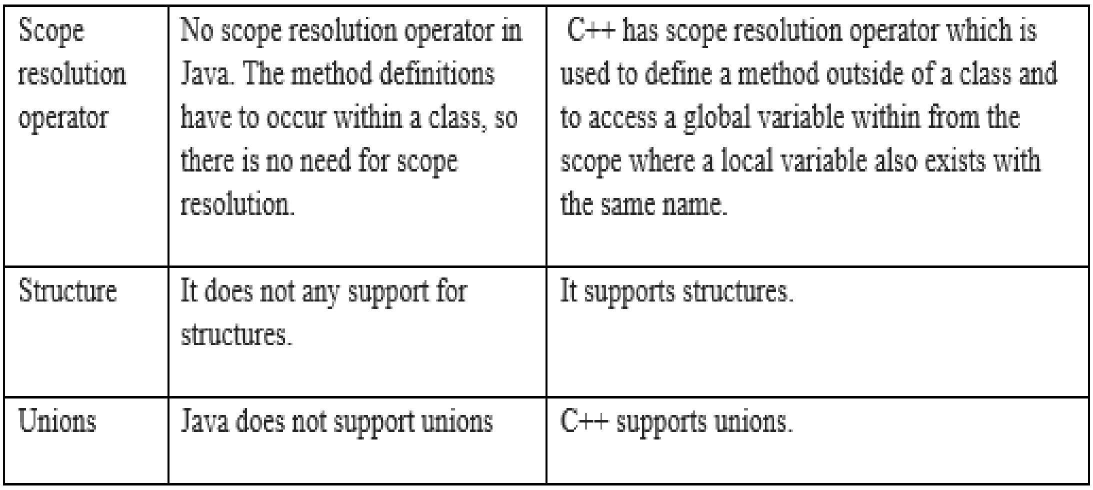
Table 1: Representation of several dissimilarities between Java and C++.



Basis of comparison	JAVA	C++
Founder	Java was developed by James Gosling at Sun Microsystems.	C++ was developed by Bjarne Stroustrup at Bell Labs since 1979 as an extension of the C language.
First Release	On May 23, 1995	In October 1985
Stable Release	Java SE 14 or JDK 14 was released on March 17, 2020.	C++17 was released in December 2017.
Official Website	oracle.com/java	isocpp.org
Influenced By:	Java was Influenced by Ada 83, Pascal, C++, C#, etc. languages.	C++ was Influenced by Influenced by Ada, ALGOL 68, C, ML, Simula, Smalltalk etc. language.
Influenced to:	Java was influenced to develop BeanShell, C#, Clojure, Groovy, Hack, J#, Kotlin, PHP, Python, Scala etc. languages.	C++ was influenced to develop C99, Java, JS++, Lua, Perl, PHP, Python, Rust, Seed 7 etc. languages.
Platform Dependency	It is platform independent, Java bytecode works on any operating System.	It is Platform dependent, should be compiled for different platform.
Portability	It can run in any OS hence it is portable.	C++ is platform dependent hence it is not portable.
Compilation	Java is both Compiled and Interpreted Language.	C++ is only Compiled Language.
Memory Management	Memory Management is System Controlled in Java.	Memory Management in C++ is Manual.
Virtual Keyword	It doesn't have Virtual Keyword.	It have Virtual Keyword.
Multiple Inheritance	It supports only single inheritance. Multiple inheritance achieved partially using interfaces.	It supports both single and multiple Inheritance.
Overloading	It supports only method overloading and doesn't allow operator overloading.	It supports both methods and operators overloading.
Pointers	It supports limited support for pointers.	It strongly supports Pointer.
Libraries	It doesn't support direct native library call, but through only Java Native Interfaces.	It supports direct system library calls, making it suitable for system-level programming.



Documentation Comment	It supports documentation comment for source code.	It doesn't support documentation comment for source code.
Thread Support	Java supports documentation comment for source code	C++ doesn't have built-in support for threads, depends on third party threading libraries.
Type	Java is only object-oriented programming language.	C++ is both a procedural and object-oriented programming language.
Input-Output mechanism	Java uses the (System class) System.in for input and System. Out for output.	C++ uses cin for input and cout for an Output operation.
Go to Keyword	Java doesn't support go to Keyword	C++ supports go to keyword.
Structures and Unions	Java doesn't support Structures and Unions.	C++ supports Structures and Unions.
Parameter Passing	Java supports only Pass by Value technique.	C++ supports both Pass by Value and pass by reference.
Global Scope	It supports no global scope.	It supports both global scope and namespace scope.
Object Management	Automatic object management with garbage collection.	It supports manual object management using new and delete.
Compiler and Interpreter	Java supports both compiler and interpreter	C++ only support compiler
Compatibility with other languages	No backward compatibility with any previous language. The syntax is influenced by C/C++.	Compatible with C source code, except some exceptional cases.
Access control and object protection	The cumbersome model encourages weak encapsulation.	Flexible model with constant protection available.
Root hierarchy	Java is a pure object-oriented programming language. That's why It follows single root hierarchy	C++ there is no such root hierarchy. C++ supports both procedural and object-oriented programming; therefore, it is called a hybrid language.
Functions & Data	All function and data exist within classes; package scope are available.	Functions and data may exist external to any class, global and namespace scopes are available



Scope resolution operator	No scope resolution operator in Java. The method definitions have to occur within a class, so there is no need for scope resolution.	C++ has scope resolution operator which is used to define a method outside of a class and to access a global variable within from the scope where a local variable also exists with the same name.
Structure	It does not any support for structures.	It supports structures.
Unions	Java does not support unions	C++ supports unions.

2. DISCUSSION

C++ vs. Java is the largest programming puzzle for many people. When it comes to creating the most complex websites and software development methods, both of these languages are recognised for their strength. Because of their history, design, and support libraries, despite the fact that both languages have object-oriented programming foundations, they were never completely object-oriented.[10] The two languages have differences on a wide range of issues. Both the languages have lots of differences and similarities between them which have been discussed in this paper. This paper helps students determining which language is best for their project according to the needs and requirements of the project.

3. CONCLUSION

Java and C++ both are really efficient programming languages used by several programmers for developing their products. Both these languages are best in their own terms with several characteristics which make them stand out from other languages. These features have been discussed in this paper. This paper discusses history of these languages, how they originated and how these grew their standard in market with time. This paper discusses execution process of code in both C++ and Java. It discusses which languages is better among C++ and java according to different situations. For instance, when it comes to system development, C++ is the way to go whereas Java, on the other hand, is a wonderful choice for desktop, web, and other applications. This paper discusses as in which language is easy to learn and implement for students along with several differences between them. As there is a large competition of languages in development field both these languages are performing excellent, which increases their implementation and use in future, thereby increasing the future scope of both these languages.

REFERENCES

[1] P. J. B.A. and K. A. Bhosale, “Research Paper on Java Interactional Development Environment Programming Tool,” *IARJSET*, 2017, doi: 10.17148/iarjset/nciarcse.2017.35.

[2] R. C. Martin, “Java and C ++ A critical comparison,” *Elements*, vol. 2, pp. 1–14, 1997.

[3] D. Gruntz, “C# and Java: The smart distinctions,” *J. Object Technol.*, vol. 1, no. 5, pp. 163–176, 2002, doi: 10.5381/jot.2002.1.5.a4.

[4] R. Dattero, J. J. Quan, and S. D. Galup, “Estimating the Value of Java and C++ Skills,” *Commun. Assoc. Inf. Syst.*, 2003, doi: 10.17705/1cais.01117.

[5] J. Souli, “C ++ Language Tutorial,” *Cogn. Technol. Work*, 2007, doi: 10.1007/978-3-540-74444-3.

[6] L. T. Journal, “Java,” pp. 3–5, 1996.

[7] S. Bose, “A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT,” *Int. J. Adv. Res. Comput. Sci.*, 2018, doi: 10.26483/ijarcs.v9i3.5978.

[8] R. Mateosian, “Java,” *IEEE Micro*. 1996, doi: 10.1109/MM.1996.502399.

[9] R. Hundt, “Loop Recognition in C ++ / Java / Go / Scala,” *Byte*, 1983.

[10] D. a. Collier and S. M. Meyer, “An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl,” *Int. J. Oper. Prod. Manag.*, 2000.