
BASICS OF JAVA PROGRAMMING LAUNGAUGE

Sanjana Singh Kushwaha*¹, Shivani Sisodiya*²

*^{1,2}Sam Global University, Bhopal, Madhya Pradesh, India.

ABSTRACT

Java is a general-purpose programming language designed to enable programmers to write code that is widely compatible and can be executed on multiple platforms without requiring recompilation. There is a high demand for Java programmers, as many popular companies seek their expertise to develop Desktop, Web, Game, and Mobile Applications. The Java programming language is relevant for both students and professionals working in the Software Development field. This paper serves as an introduction to the Java programming language, explaining its popularity compared to other languages and highlighting some of its key features. Additionally, it includes ten reasons to learn Java and explores the various applications of the Java programming language.

Keywords: Java Programming Language, Platform Independent, Write Once And Run Anywhere (WORA), Multithreaded, Java Virtual Machine (JVM), Bytecode.

I. INTRODUCTION

The Java programming language was developed by James Gosling in 1995. Java has become extensively popular as a class-based, object-oriented, and high-level programming language. It is designed to enable the "write once, run anywhere" (WORA) principle, which allows compiled Java code to run on any platform that supports Java without the need for further compilation. Java is known for its simplicity, portability, and platform independence, making it a popular choice for developing a wide range of applications, including mobile apps, web apps, desktop apps, and games. One of the key features of Java is its platform independence, as the code is compiled into bytecode, which can run on any platform that supports the Java Virtual Machine (JVM). This allows developers to write code once and run it on any platform without having to worry about platform-specific details. In addition, Java is widely used in industry and academia, with many organizations relying on Java for critical business applications.

II. WHY JAVA IS POPULAR COMPARED TO OTHER PROGRAMMING LANGUAGES?

One of the primary reasons for Java's widespread adoption is its platform independence. Java bytecode can Run on different systems, as it is platform-independent. The Java compiler converts Java code into bytecode. Platform independence means that a program can run on any platform or operating system, regardless of its underlying configuration, thanks to the Java Virtual Machine. Java adheres to the WORA principle, which stands for "Write Once, Run Anywhere." Additionally, Java is well-known for its security features, which protect against intrusion, replication, and virus issues. Java supports multithreading, allowing developers to create multiple threads within a single process that operate independently and concurrently. Furthermore, due to its high performance, Java has a wide range of applications in developing mobile applications, web applications, desktop applications, and games, which contributes to its popularity.

III. FEATURES OF JAVA PROGRAMMING LANGUAGE

Java is an extensively used programming language known for its performance, platform independence, and security. It is useful for various applications, including back-end development, Android development, desktop computing, game development, and numerical computing. This paper discusses the top features of the Java programming language.

1. Object Oriented

All entities in Java are represented as objects. Anyone can design software by integrating various forms of elements that include both functions and data. The object-oriented programming (OOP) model in Java has contributed to its popularity in developing complex software systems, especially in enterprise environments where modularity and scalability are fundamental considerations.

2. Simple

Java is easy to learn because its syntax is simple, clean, and straightforward. According to Sun Microsystems, the syntax is based on C, which makes it familiar to programmers who have experience with C or similar languages. Additionally, Java offers a vast library of built-in classes and methods that provide many common functionalities

that developers need when building applications. Another reason for its simplicity is platform independence; developers can write code once and run it on any platform that supports Java. Overall, Java's ease of use makes it a popular choice for both beginners and experienced developers, contributing to the language's widespread adoption and success.

3. Secured

Java is best known for its security features, as it has developed a system that is free from common vulnerabilities. It runs inside a virtual machine, which helps isolate it from the underlying hardware. Additionally, Java utilizes a class loader that enhances security by separating class packages from imported network resources and the original operating system. To ensure secure access to appropriate objects, the bytecode verifier checks the code for any violations of security rules. Meanwhile, the security manager determines which resources a class can access within the local environment.

4. Platform Independent

The Java platform is unique compared to other platforms because it is built on software-based technology that operates above hardware-based platforms. The platform consists of two main components:

1. Runtime Environment: This runs on top of a computer's operating system and provides the class libraries and other resources needed to execute a Java program.
2. API (Application Programming Interface): This offers a straightforward way to access and share data within and between applications.

5. Robust

Java is regarded as a secure and resilient programming language, primarily due to its efficient memory management system. One significant advantage of Java is the absence of pointers, which helps reduce security vulnerabilities. Furthermore, Java utilizes automatic garbage collection through the Java Virtual Machine, effectively removing unused objects within a Java application. These features contribute to making Java a robust choice for developers.

6. Portable

Java is classified as a portable programming language because code written in Java is compiled into bytecode. This bytecode can then be executed using a Java Virtual Machine (JVM), which functions similarly to an interpreter by reading the compiled bytecode and running it.

Java is considered more portable than C and C++ for several reasons. In C and C++, compiling code can take a significant amount of time, often hours, while Java allows for quicker execution of bytecode through the JVM. This is due to Java's "write once, run everywhere" (WORA) feature, which means that code can be written once and run on any platform that supports the JVM without requiring recompilation.

7. Architecture Neutral

Java is designed with the "write once, run anywhere" (WORA) philosophy in mind. Its architecture is platform-independent because it effectively hides the complex details of the underlying system on which Java code runs. There are multiple Java Runtime Environment (JRE) implementations available, tailored for various operating systems.

8. Dynamic

Bytecode is the reason Java is classified as a dynamic language. The code is generated for a specific platform but can run on any platform. The class file is loaded only at runtime, which means that all actions that take place during that time are considered dynamic.

9. Interpreted and Compiled

Java is a programming language that can be executed using both interpretation and compilation processes. The question of whether Java is compiled or interpreted can be somewhat unclear. The process begins by compiling the Java source code into a binary bytecode using the Java compiler. This bytecode is then interpreted by the software-based Java Virtual Machine (JVM). Therefore, Java is considered to be both compiled and interpreted.

10. High-Performance

The growing interest in Java for High-Performance Computing (HPC) is primarily due to its appealing features for programming multi-core cluster architectures. Java offers built-in support for networking and multithreading,

which are essential in this area. Moreover, the continuous improvement in the performance of the Java Virtual Machine (JVM) has enhanced its appeal even further in the HPC context.

IV. 10 REASONS TO LEARN JAVA PROGRAMMING LANGUAGE

1. Popularity and High Salary

The demand for Java is significant in today's job market. Its popularity has led to around 9 million developers using Java, which runs on approximately 7 billion devices worldwide. Due to this high demand, Java developers often command attractive salaries, making them highly sought after for many of the industry's top-paying positions. As shown in Figure 6, the average salary for a Java Developer ranges from \$44,989 to \$119,428 per year.

2. Easy to Learn

Java is simple and easy to learn. Its straightforward syntax allows for quick understanding, making it easy to write and maintain code. Furthermore, Java adheres to the WORA principle, meaning developers can write their code once and run it anywhere.

3. Large Community

Java is a widely used programming language that has maintained its popularity for over two decades. This popularity has attracted a substantial number of developers who actively use and contribute to the language. As an open-source language, its source code is available for anyone to view and modify. The "write once, run anywhere" (WORA) feature makes it easy to develop and deploy applications across different platforms, including Windows, Linux, and macOS. Additionally, many large corporations, such as Google, Amazon, and Oracle, rely on Java in their products and services.

4. Abundant API

The Programming Interface, commonly known as an API, comprises a diverse range of Java classes, packages, and interfaces. This system is designed to facilitate development without requiring familiarity with internal implementations. The API is primarily divided into three categories:

1. Core Java APIs: These include various technologies, such as Android (from Google), SE (OpenJDK and Oracle), and Micro EJ. These fundamental Java packages are essential for effective programming in Java, as they provide the foundation upon which developers build their applications.
2. Optional Java APIs: These APIs are designed to handle scenarios where it may be necessary to indicate the absence of a result. Using `null` can lead to potential issues, so it is advisable to avoid `null` values for Optional variables. Striving for optimal use of Optional types should always be a goal.
3. Unofficial APIs: These are created through the reverse engineering of sanctioned APIs by utilizing the front end of a service.

Overall, understanding and effectively using these different types of APIs is crucial for efficient Java programming.

5. Multi Open-source Libraries

Numerous inventors contribute to open-source libraries, which helps to make a rich collection of libraries. With its modular design, the inventors can use existing libraries to break common programming problems. Concerning cost savings, open-source libraries are frequently free to use. Thus, inventors can save on licensing freights and other costs associated with personal libraries.

6. Powerful Development Tools

Java has several powerful Integrated Development Environments (IDEs) such as Eclipse, NetBeans, and IntelliJ IDEA. The success of Java is largely attributed to these IDEs, which offer a variety of features including debugging, syntax highlighting, code completion, language support, automated refactoring, and more. Rendering in Java is made more feasible and flexible thanks to these tools.

7. Free of Cost

Java's widespread use among independent programmers can be attributed to its accessibility through the Oracle Binary Code License (BCL) at no cost. While practical use of Java does require payment, it is available for free in development and testing environments.

8. Platform Independent

The compiler converts Java source code into bytecode that can run on any platform with the help of the Java Virtual Machine (JVM). Java is referred to as a WORA (Write Once, Run Anywhere) language due to its platform-agnostic capability, meaning it can be written once and executed on any system.

9. Great Documentation Support

Javadoc is a leading tool for creating documentation for Java, providing exceptional support for the language's needs. The API documentation is generated in HTML format using Java source code, making Javadoc an invaluable resource during Java programming. This ensures that users can easily understand the concepts and structures developed in the code.

10. Versatile

Java is utilized across various platforms to develop a wide range of applications, including those for the web, mobile devices, desktops, and games. Additionally, Java offers numerous features such as strong support for error handling, extensive security measures, platform independence, and network-centric design. It demonstrates a high degree of flexibility.

V. CONCLUSION

Java is a high-level, object-oriented programming language that has gained immense popularity since its creation in 1995. It is applicable to a wide range of applications, including mobile apps, web applications, desktop applications, and games.

Java is known for its simplicity, portability, security, robustness, multi-threading capability, and platform independence, making it an ideal language for developing complex and high-performance software systems. Its popularity can be attributed to several factors, including ease of use, platform independence, a large community, open-source libraries, and the "write once, run anywhere" philosophy.

Additionally, Java has a rich ecosystem of tools and frameworks that support various development workflows and infrastructures. The demand for Java developers remains consistently high, providing excellent career opportunities and competitive salaries. Overall, Java's versatility and reliability make it one of the most widely used programming languages worldwide, and it continues to be a popular choice for developers in the future.

VI. REFERENCES

- [1] Singh, T. (2012). New Learning Methodology for Student of Java Programming Language. International Journal of Engineering Research and Development, 17-19
- [2] H. Kopetz and G. Bauer, "The time-triggered architecture," Proceedings of the IEEE, vol. 91, pp. 112-126, Jan 2003
- [3] N. Burri, R. Flury, S. Nellen, B. Sigg, P. Sommer, and R. Wattenhofer, "Yeti: an eclipse plug-in for tinyos 2.1," in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, pp. 295-296, ACM, 2009
- [4] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al., "Tinyos: An operating system for sensor networks," in Ambient intelligence, pp. 115-148, Springer, 2005.
- [5] Comparison of integrated development environment (ide) debugging tools: eclipse vs net beans. *1 mrs. kavita s., *2ms. sindhu s., (july 2015).
- [6] Yin Liu, A. M. (2010). Static Information Flow Analysis with Handling of Implicit Flows and A Study on Effects of Explicit Flows vs Implicit Flows. Software Maintenance and Reengineering (CSMR), 2010 14th European Conference (pp. 1-10). Madrid, Spain: IEEE
- [7] Rafael H. Bordini, L. B. (2006). A Survey of Programming Languages and Platforms for Multi-Agent Systems. Informatica, 33-44.