

An Overview On Java Programming

Swapnil Raj

SOEIT, Sanskriti University, Mathura, Uttar Pradesh, India

Email Id- swapnil.cse@sanskriti.edu.in

ABSTRACT: Coding and programming are all around us, and will continue to grow in popularity in the future. Learning to programme is one method to break into the IT industry and java is one of most famous programming language. Java is a platform as well as a programming language. Java is a high-level programming language that is also robust, object-oriented, and secure. Java was created in 1995 by Sun Microsystems (which is now a division of Oracle). The founder of Java, James Gosling, is regarded as the "Father of Java." It was known as Oak before Java. Because Oak was already a recognized business, James Gosling and his colleagues decided to alter the name to Java. In this paper we will studies about what is java, java features, java application and element of java.

KEYWORDS: Casting, Framework , Java , Object Oriented, Variable.

1. INTRODUCTION

James Gosling and his Sun Microsystems team proposed the Java programming language in 1991, and it was launched in 1995. Java's most notable characteristic is that it is platform-agnostic, i.e., it possesses WORA qualities (Write Once, Run Anywhere). OAK was the name given to Java when it was initially conceived. OAK was first presented as a programming language that serves as a platform for connecting appliances such as VCRs, TVs, and other devices. In 2009-10, Oracle Corporation purchased Sun Microsystems and became the sole owner of Java. Java is a statically typed compiled language, which means that variables must be defined before values may be assigned. Java applications are quicker than Python programmes, although they are slower than C++ ones. Java is used by Airbnb, Uber, LinkedIn, Pinterest, Groupon, Spotify, Eclipse, Hadoop, and many more companies. Java is still used by large companies like as Infosys, TCS, Wipro, HCL Tech, Naukri, Jabong, Myntra, Flipkart, Trivago, ibibo, and others. Java is a class-based, object-oriented programming language with a focus on minimising implementation dependencies. A compiled general-purpose programming language designed for developers to write once and execute anywhere. All systems that support Java can run Java code. Java programmes are compiled into byte code that may be executed on any Java Virtual Machine. Java's syntax is comparable to that of C/C++[1].

The history of Java is fascinating. It is a programming language that was first introduced in 1991. The Java language was created in 1991 by a group of Sun developers known as the Green team, which included James Gosling, Mike Sheridan, and Patrick Naughton. Sun Microsystems published Java 1.0, the first public implementation, in 1996. It offers no-cost run-times on a variety of platforms. Arthur Van Hoff rewrote the Java1.0 compiler in Java to ensure full adherence to its requirements[2]. With the release of Java 2, new versions included several settings for various systems[3].

Sun Microsystems approached the ISO standards organisation in 1997 and later standardised Java, but it quickly dropped out. Sun used to make most of their Java implementations accessible for free, despite the fact that they were proprietary software. Java brought substantial money for Sun by selling licences for specific products like the Java Enterprise System[4]. Sun released much of their Java virtual machine as free, open-source software on November 13, 2006. Sun completed the process on May 8, 2007, by making all of its JVM core code accessible under open-source licence conditions. Simple, resilient, secure, high speed, portable, multi-threaded, interpreted, dynamic, and other concepts guided the development of Java. James Gosling, sometimes known as the Father of Java, created Java in 1995[5]. Java is being utilised in mobile devices, internet programming, gaming, e-business, and other applications[6].

Java is an object-oriented programming language. Everything in Java is linked to a class or an object. In Java, the term "class" is used to declare a class .In Java, an object is used to access class members[7].

When creating a new object, the term "new" is used. There can only be one main class and one main function in a Java application. In Java, a function Object() { [native code] } has the same name as the class. Even if the programmer does not define a function Object() { [native code] }, a default function Object() { [native

code] } is imposed whenever an object is created in both languages. Because the function Object() { [native code] } is not included in the example, it is automatically added during compilation. In Java, a static variable (also known as a class variable) is a variable that is shared by all instances in a class and is defined with the keyword "static." Instance variables are defined outside of the method and are not static. A method contains a local variable.

1.1 Features of Java

- **Object-oriented:** Java is an object-oriented programming language. When dealing with real-world applications, OOP (Object-Oriented Programming) comes in handy. Incorporating inheritance, the combination of polymorphism, abstraction, and encapsulation results in An OOP is a programme.
- **Platform Independent:** When a programme is compiled, it is converted into platform-independent byte code, which is then run by a Java Virtual Machine (JVM). The usage of JVM makes Java programming platform agnostic, since the same application may be run on various systems if JVM is installed.
- **Secured:** Java applications are safe because they employ their own runtime environment, the JVM. Java has built-in safety features such as typechecking at compile time and runtime checking. Java is also well-known for its security features. Java is also devoid of pointers, which promotes security.
- **Robust:** Java provides excellent memory management and immediately deletes things that are no longer in use. Exception handling and other sorts of checking mechanisms are included in Java. Java's robustness is due to all of these traits.
- **Multi-thread:** Multi-threading capabilities are integrated into Java. It assists in the development of highly dynamic and responsive apps that handle several tasks at once. Multi Threads share a shared memory region, which improves performance and capability.

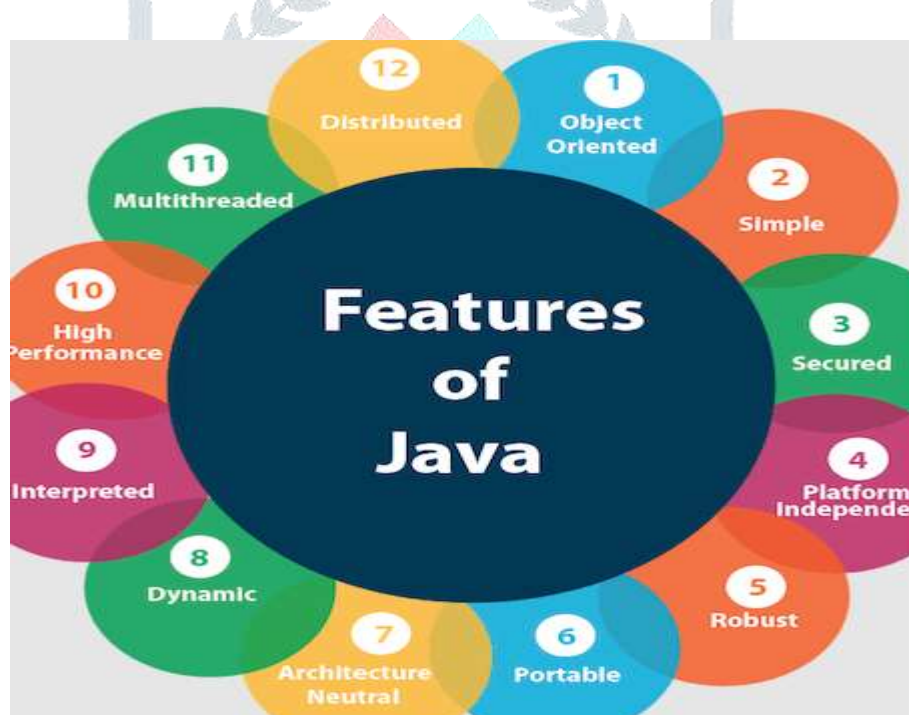


Figure 1: Diagrammatic Representation of Features of Java [JAVATPOINT]

1.2 Applications of Java

In the actual world, Java is used in a variety of disciplines. The group did some study on Java's applications. Below is a quick rundown of some of Java's most frequent applications.

- **Android Applications:** The official programming language for developing Android smartphone apps is Java. Although Dart may be used to develop mobile applications, Java is the most popular. The Dalvik Virtual Machine is a customized virtual machine for Android that runs the Java byte code that was built (DVM). Java is supported by application development technologies such as Android Studio

and Kotlin. With Android application development, the Java OOP concept provides improved security, simplicity, and effectiveness.

- **Desktop GUI Applications:** The designed desktop application may be written in Java. To create GUI applications, tools like Abstract Windowing Toolkit (AWT), JavaFX, and Swing are utilized.
- **Web-based Applications:** With the aid of servlets, struts, and JSP (Java Server Pages), Java is used to develop web applications. Java offers simple code and excellent security, making it ideal for developing applications in the fields of health, social security, education, and insurance. Broadleaf and other open-source ecommerce systems let Java developers create e-commerce web apps.
- **Cloud-based Applications:** Cloud computing results in a low-cost IT infrastructure. It features pay-as-you-go pricing and on-demand delivery of IT resources through the internet, including storage, servers, databases, networking, and applications. Java contains characteristics like SaaS (Software-as-a-Service), IaaS (Infrastructure-as-a-Service), and PaaS (Platform-as-a-Service) that help with development (Platform-as-a-Service)
- **Big Data Technologies:** Java is used to analyse Big Data because it is quick, dependable, and resilient. Java's characteristics, such as Automatic Garbage Collection and robust memory management, make it ideal for Big Data applications. Java uses frameworks like Apache Mahout, Apache Spark, and Java JFreechart to engage in Big Data. Because of its extensive roots in open source communities, Java is a better fit for all of these technologies. Apache Hadoop, Apache Spark, Apache Mahout, and other big data technologies are Java sub-projects.
- **Mobile App Development:** The Java programming language is widely regarded as the standard for mobile app development. The majority of Android applications are written in Java. Android Studio, the most popular android app development IDE, also uses Java to build android apps. As a result, if you are already comfortable with Java, developing Android applications will be considerably easier. Java is used to create the most popular Android apps, such as Spotify and Twitter.

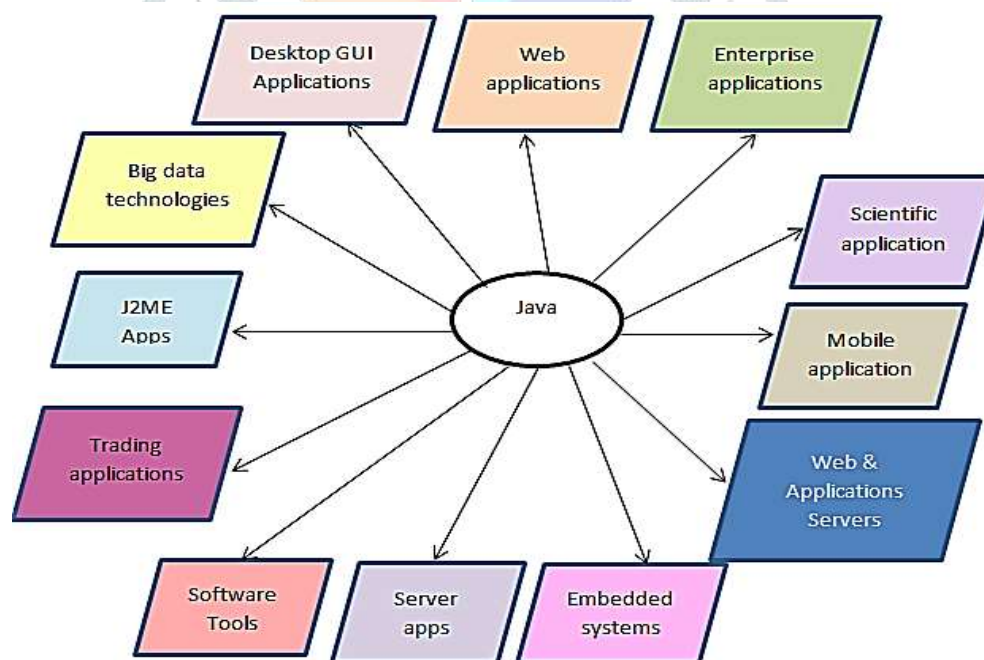


Figure 2: Diagrammatic Representation of Application of Java [SOFTWARETESTINGHELP]

1.3 Elements Of Java

- 1.3.1. **Variables:** Variables in Java, on the other hand, must be declared ahead of time, along with their type. Although a variable can be declared before it is assigned a value, it must confirm its datatype at the time of declaration. The value of a variable can be changed, but it must be of the same datatype. The term "final" in Java helps to prevent the value from being overwritten. A comma-separated list can be used to declare several variables of the same type. Local variables, static variables (class variables), and instance variables are the three types of variables in Java.

1.3.2. Casting: In java there is two types of casting

- The process of converting a smaller data type to a bigger data type, such as byte to short or int to double, is known as implicit or automated type casting. If variable an is a short and the conversion is from short to int, int b=a;
- typecasting (explicit, narrowing, or manual) It is the conversion of a bigger data type to a smaller data type, such as a double to a short or an int to a byte. If variable x is a float and the conversion is from float to int, then int y=(int) x; Casting types in Java

1.3.3. Operators: Java includes a variety of operators, some of which are comparable to those found in Python. Unlike Python, Java provides arithmetic operations like ++ and — for increment and decrement. There is no identification and membership operator in Java. In Java, the logical operators &&, ||, and ! stand for and, or, not, respectively.

1.3.4. Print Function: The syntax of Java is more organised. The “+” operator is used to concatenate variables in a print statement. System.out.print() returns a list of strings or references. System.out.println(); is identical to the above, except it advances the cursor to the next line after printing.

1.3.5. Comments: For single-line comments, Java employs the “//” (two forward slashes) character; everything following the “//” is considered a comment. “/* ” is used at the beginning of a multi-line comment, while “ */ ” is used at the conclusion. Between them, any text or code is disregarded.

1.3.6. Separators: To separate two lines in Java, use the semicolon ";" character. If “;” is not inserted after each statement, an error will occur. Programmers are urged to double-check the “;” since even a single “;” can cause a huge mistake.

2. LITERATURE REVIEW

[8] Propose that AspectJ^ô is a simple and useful aspect-oriented Java^ô extension. AspectJ supports the modular implementation of a variety of crosscutting concerns with just a few additional constructs. Join points are well-defined points in the execution of the programme in AspectJ's dynamic join point model; pointcuts are collections of join points; advice are special method-like constructs that can be attached to pointcuts; and aspects are modular units of crosscutting implementation, consisting of pointcuts, advice, and ordinary Java member declarations. The code written in AspectJ is compiled into normal Java bytecode. Simple additions to current Java development environments enable browsing of the crosscutting structure of aspects in the same manner that the inheritance structure of classes is browsed. AspectJ is strong, and applications created with it are simple to comprehend, as demonstrated by several examples.

[9] Propose that JPaX can track the execution of a Java application and ensure that it complies with a set of temporal logic-based characteristics supplied by the user. JPaX can also do analysis. Concurrency faults, such as deadlocks and data races, are detected by this software. The study of concurrency There is no requirement for a user-supplied specification. The utility allows for automatic instrumentation of a system. The bytecode of a programme, which when run, emits an event stream, the execution trace, to an output device. observer. The observer distributes the incoming event stream to a group of observer processes, each of which performs a different task. conducting a particular analysis, such as the verification of temporal logic, the deadlock analysis and the data race analysis are two types of analyses. The user can construct temporal logic requirements in the Maude rewriting logic, which is a high-speed rewriting system for equational logic that has been expanded to include executable temporal logic. After then, the Maude rewriting engine is turned on as an event-driven monitoring process. Temporal requirements can also be converted into efficient automata that check the event stream. JPaX can be used to gather more information about programme executions during programme testing, and it may also be used to survey safety-critical systems during operation.

[10] Propose that Java Statistical Analysis Tool (JSAT) is a Machine Learning library written in pure Java. It works to fill a void in the Java ecosystem for a general purpose library that is relatively high performance and exible, which is not adequately fulfilled and Java-ML. Almost all of the algorithms are independently implemented using an Object-Oriented framework.

3. DISCUSSION

Java is one of the most extensively used programming languages on the planet. Java is a multithreaded computer language that is simple, general-purpose, object-oriented, interpreted, resilient, secure, architecture-neutral, portable, and high-performance. It is designed to allow application developers to "write once, run anywhere" (WORA), which means that code written for one platform does not need to be recompiled for another.

Java is a computer language as well as a platform. Java is an object-oriented programming language that is high-level, robust, and secure. A platform is defined as any hardware or software environment in which a programme operates. Java is referred to be a platform since it has its own runtime environment (JRE) and API. All source code in the Java programming language is first written in plain text files with the .java extension. The javac compiler subsequently compiles the source files into .class files. A .class file does not contain native code for your processor; instead, it includes bytecodes, which are the Java Virtual Machine's machine language (Java VM). The java launcher utility then executes your programme on a Java Virtual Machine instance. The same .class files can operate on Microsoft Windows, the Solaris™ Operating System (Solaris OS), Linux, or Mac OS X since the Java Virtual Machine is accessible on so many other operating systems.

4. CONCLUSION

In Java set of curly brackets define a scope. A variable declared inside a class but outside any function is called class variable scope. It can be accessed and modified from anywhere inside the class. If a variable is declared inside a method, then it is called local or method variable scope. It can be accessed and modified inside the specific method only. If a variable is declared inside the block scope, which is defined by curly brackets, the variable will only be accessible and changeable within the block scope, regardless of whether the block of quotations exists on its own or is part of the if, while, or for statements. Whether the block of quotes exists on its own or is part of the if, while, or for statements, if a variable is declared inside the block scope, which is specified by curly brackets, the variable will only be accessible and modifiable inside the block scope.

The terms function and method are interchangeable in Java. In Java, there are several predefined methods. When declaring a method, it must have the access modifier static or not, the return type, and the name of the method. The return value must be of the same data type as the return type that was defined when the method was called. When there is no return value, the return type "void" is used. The arguments must be supplied in the same order as the method's corresponding parameter. Because it belongs to the class, a static method may be invoked without first generating an object. Only a static variable and a static block may be accessed by the static method, but the non-static method can access the static variable, non-static variable, static block, and non-static block. The function Object () { [native code] } in Java is not the same as the method in Java. There can't be any return value in a Java function Object () {[native code]}. A function Object () {[native code]} can't be abstract, static, or final, and it can't be overridden, either. Overloading in Java will be looked into in more depth in future articles. As you have seen the present and future scope and career opportunities in the field of Java programming, you must be convinced that learning Java programming totally worth your time and effort if you are willing to work in the IT sector.

REFERENCES

- [1] M. A. Marfai *et al.*, "Natural hazards in Central Java Province, Indonesia: An overview," *Environ. Geol.*, 2008, doi: 10.1007/s00254-007-1169-9.
- [2] A. M. Waterhouse, J. B. Procter, D. M. A. Martin, M. Clamp, and G. J. Barton, "Jalview Version 2-A multiple sequence alignment editor and analysis workbench," *Bioinformatics*, 2009, doi: 10.1093/bioinformatics/btp033.
- [3] B. Hopkins and R. Antony, *Bluetooth for Java*. 2003.
- [4] F. IV, R. YG, O. OI, D. CR, H. BM, and M. RC, "Experience in the application of Java Technologies in telemedicine.," *eHealth Int.*, 2002, doi: 10.1186/1476-3591-1-3.
- [5] D. Octeau, S. Jha, and P. McDaniel, "Retargeting Android applications to Java bytecode," 2012, doi: 10.1145/2393596.2393600.
- [6] T. Frickey and A. Lupas, "CLANS: A Java application for visualizing protein families based on pairwise similarity," *Bioinformatics*, 2004, doi: 10.1093/bioinformatics/bth444.

- [7] M. Castañón-Puga, J. M. Flores-Parra, J. R. Castro, C. Gaxiola-Pacheco, and L. E. Palafox-Maestre, "Building Java intelligent applications data mining for Java Type-2 Fuzzy Inference Systems," 2015, doi: 10.1016/j.procs.2015.05.401.
- [8] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold, "An overview of AspectJ," 2001, doi: 10.1007/3-540-45337-7_18.
- [9] K. Havelund and G. Roşu, "An Overview of the Runtime Verification Tool Java PathExplorer," 2004, doi: 10.1023/B:FORM.0000017721.39909.4b.
- [10] E. Raff, "JSAT: Java statistical analysis tool, a library for machine learning," *J. Mach. Learn. Res.*, 2017.

