

Website

<https://www.curiousinfotecholutions.com>

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Java Programming

Day-1



[Java - Features \(Class, Object, etc\)](#)

[Java - What, Why, History, Version?](#)

Email id:
info@curiousinfotecholutions.com

Syllabus Day-1

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- What is Java?
- Why use Java?
- What is History of Java?
- Brief Java Version History.
- What are Features of Java?
- Explore C++ vs Java.

Compilation Flow of Java Program

Parameters used in Java Program



What is Java?

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

PDF

- Java is a popular programming language, created in 1995.
- It is owned by Oracle, and more than **3 billion** devices run Java.
- Java is an object-oriented, class-based, concurrent, secured and general-purpose computer-programming language. It is a widely used robust technology.
- It is used for:
 - ✓ Mobile applications (especially Android apps)
 - ✓ Desktop applications
 - ✓ Web applications
 - ✓ Web servers and application servers
 - ✓ Games
 - ✓ Database connection
 - ✓ And much, much more!





Why Use Java?



- Java works on **different platforms** (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world.
- It is easy to **learn and simple to use**.
- It is **open-source and free**.
- It is **secure, fast and powerful**.
- It has a huge community support (tens of millions of developers)
- Java is an object-oriented language which gives a **clear structure** to programs and allows **code to be reused**, lowering development costs.
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa.

History of Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. The small team of sun engineers called Green Team.
- Initially it was designed for small, embedded systems in electronic appliances like set-top boxes.
- Firstly, it was called "Greentalk" by James Gosling, and the file extension was .gt.
- After that, it was called Oak and was developed as a part of the Green project.

Why Oak? Oak is a symbol of strength and chosen as a national tree of many countries like the U.S.A., France, Germany, Romania, etc.

History of Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies.
- The team gathered to choose a new name. The suggested words were "dynamic", "revolutionary", "Silk", "jolt", "DNA", etc. They wanted something that reflected the essence of the technology: revolutionary, dynamic, lively, cool, unique, and easy to spell, and fun to say.
- According to James Gosling, "Java was one of the top choices along with Silk". Since Java was so unique, most of the team members preferred Java than other names.

History of Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Java is an island in Indonesia where the first coffee was produced (called **Java coffee**). It is a kind of espresso **bean**. Java name was chosen by James Gosling while having a **cup of coffee** nearby his office. **Java is just a name**, not an acronym.
- Initially developed by James Gosling at **Sun Microsystems** (which is now a subsidiary of **Oracle Corporation**)
- **JDK 1.0 was released on January 23, 1996**. After the first release of Java, there have been many additional features added to the language. Now Java is being used in **Windows applications, Web applications, enterprise applications, mobile applications, cards, etc.** Each new version adds new features in Java and released in 1995.

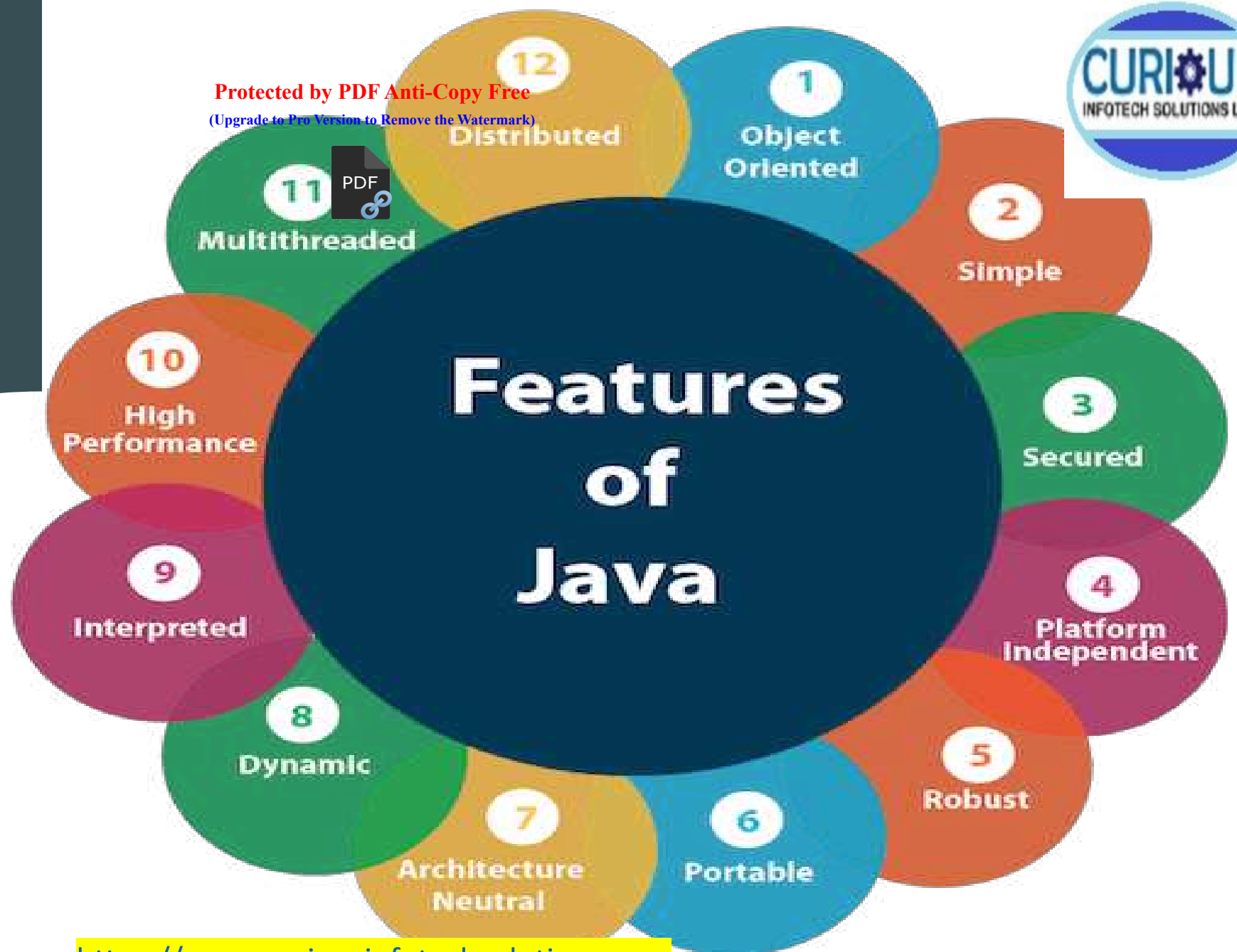
Java Version History

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan 1996)
3. JDK 1.1 (19th Feb 1997)
4. J2SE 1.2 (8th Dec 1998)
5. J2SE 1.3 (8th May 2000)
6. J2SE 1.4 (6th Feb 2002)
7. J2SE 5.0 (30th Sep 2004)
8. Java SE 6 (11th Dec 2006)
9. Java SE 7 (28th July 2011)
10. Java SE 8 (18th Mar 2014)
11. Java SE 9 (21st Sep 2017)
12. Java SE 10 (20th Mar 2018)
13. Java SE 11 (September 2018)
14. Java SE 12 (March 2019)
15. Java SE 13 (September 2019)
16. Java SE 14 (Mar 2020)
17. Java SE 15 (September 2020)
18. Java SE 16 (Mar 2021)
19. Java SE 17 (September 2021)
20. Java SE 18 (to be released by March 2022)

Features of Java



Java is an Object-oriented Programming Language

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporate both data and behavior.
- Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.
- Basic concepts of OOPs are:
 - Object
 - Class
 - Inheritance
 - Polymorphism
 - Abstraction
 - Encapsulation ▲

Java is simple.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Easy to learn, and its syntax is simple, clean and easy to understand.
- According to Sun Microsystem, Java language is a simple programming language because:
 - Java syntax is based on C++ (so easier for programmers to learn it after C++).
 - Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
 - There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Java is platform independent.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a **write once, run anywhere** (WORA) language. A platform is the hardware or software environment in which a program runs.
- There are two types of platforms software-based and hardware-based. Java provides a software-based platform.
- The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on top of other hardware-based platforms. It has two components:
 1. Runtime Environment
 2. API (Application Programming Interface)



Java is best known for its security.

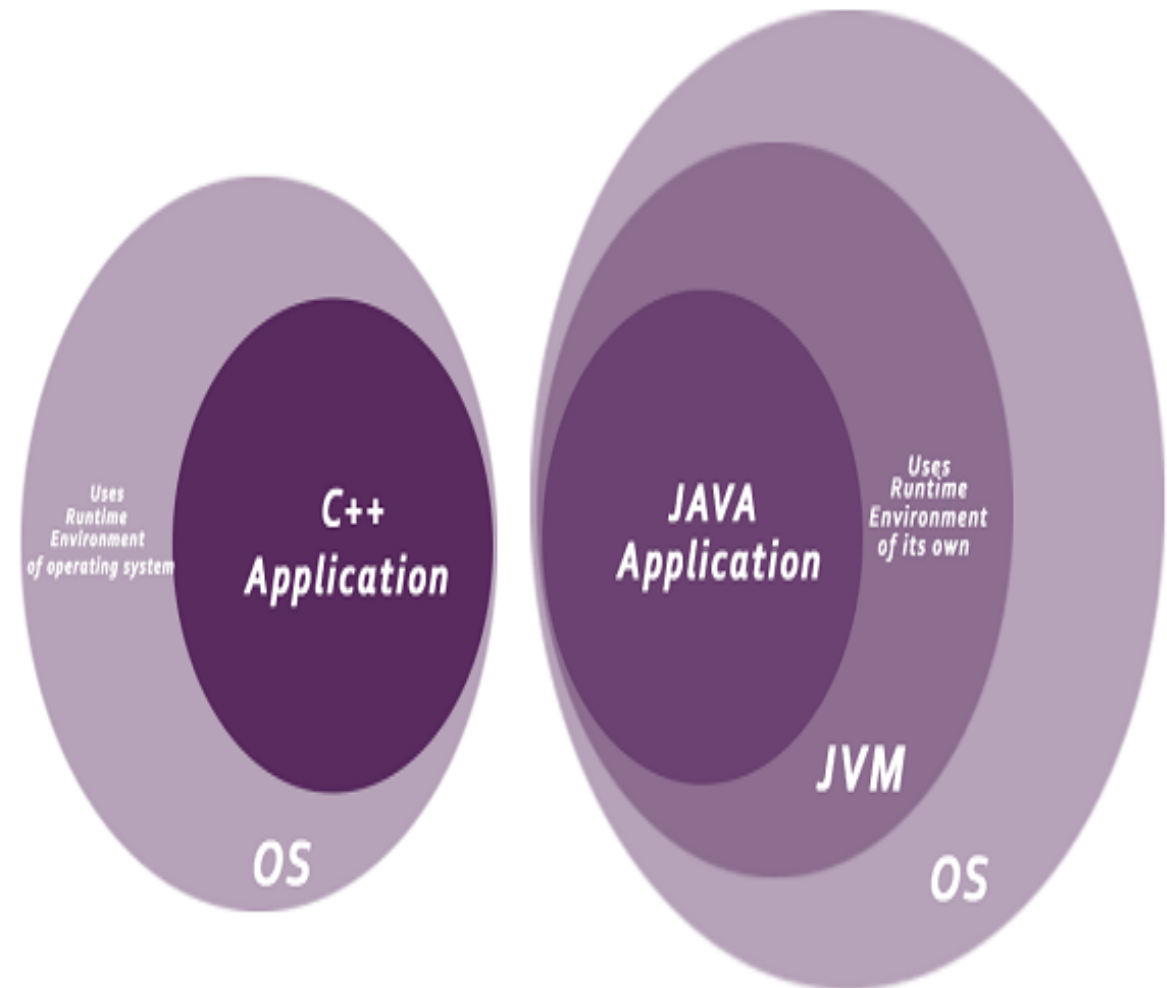
Protected by PDF Annotator
(Upgrade to Pro Version to Remove the Watermark)



- With Java, we can develop virtual machines because:

- **No explicit pointer**
- **Java Programs run inside a virtual machine**

- **Bytecode Verifier:** It checks the code fragments for illegal code that can violate access rights to objects.
- **Security Manager:** It determines what resources a class can access such as reading and writing to the local disk.
- **ClassLoader:** Classloader in Java is a part of the Java Runtime Environment (JRE) which is used to load Java classes into the Java Virtual Machine dynamically.



Contd..

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Java is Robust.

- The English meaning of Robust is strong. Java is robust because:
- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- Java provides automatic garbage collection which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.

Java has Architecture-neutral.

- Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.
- In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java.

Contd..

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Java is Portable Language.

- Means, it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

Java has High-performance.

- Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g., C++). Java is an interpreted language that is why it is slower than compiled languages, e.g., C, C++, etc.

Java is Distributed Language.

- Java is distributed because it facilitates users to create distributed applications in Java. RMI (Remote Method Invocation) and EJB (enterprise java bean) are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.

Contd..

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Java is Multi-threaded Language

- A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.

Java is Dynamic Language

- Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

C++ vs Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Comparison Index	C++	Java
Platform-independent	C++ is platform-dependent.	Java is platform-independent.
Mainly used for	C++ is mainly used for system programming.	Java is mainly used for application programming. It is widely used in Windows-based, web-based, enterprise, and mobile applications.
Design Goal	C++ was designed for systems and applications programming. It was an extension of the C programming language.	Java was designed and created as an interpreter for printing systems but later extended as a support network computing. It was designed to be easy to use and accessible to a broader audience.
Goto	C++ supports the <u>goto</u> statement.	Java doesn't support the goto statement.

Contd.. C++ vs Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Multiple inheritance	C++ supports multiple inheritance.	Java doesn't support multiple inheritance through class. It can be achieved by using interfaces in java .
Operator Overloading	C++ supports operator overloading .	Java doesn't support operator overloading.
Pointers	C++ supports pointers . You can write a pointer program in C++.	Java supports pointer internally. However, you can't write the pointer program in java. It means java has restricted pointer support in java.

Contd..C++ vs Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Compiler and Interpreter	C++ uses compiler only. C++ is compiled and run using the compiler which converts source code into machine code so, C++ is platform dependent.	Java uses both compiler and interpreter. Java source code is converted into bytecode at compilation time. The interpreter executes this bytecode at runtime and produces output. Java is interpreted that is why it is platform-independent.
Call by Value and Call by reference	C++ supports both call by value and call by reference.	Java supports call by value only. There is no call by reference in java.
Structure and Union	C++ supports structures and unions.	Java doesn't support structures and unions.

Contd.. C++ vs Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Thread Support	C++ doesn't have built-in support for threads. It relies on third-party libraries for thread support.	Java has built-in thread support.
Documentation comment	C++ doesn't support documentation comments.	Java supports documentation comment (<code>/** ... */</code>) to create documentation for java source code.
Virtual Keyword	C++ supports virtual keyword so that we can decide whether or not to override a function.	Java has no virtual keyword. We can override all non-static methods by default. In other words, non-static methods are virtual by default.
unsigned right shift >>>	C++ doesn't support >>> operator.	Java supports unsigned right shift >>> operator that fills zero at the top for the negative numbers. For positive numbers, it works same like >> operator.

Contd.. C++ vs Java

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Inheritance Tree	C++ always creates a new inheritance tree.	Java always uses a single inheritance tree because all classes are the child of the Object class in Java. The Object class is the root of the <u>inheritance</u> tree in java.
Hardware	C++ is nearer to hardware.	Java is not so interactive with hardware.
Object-oriented	C++ is an object-oriented language. However, in the C language, a single root hierarchy is not possible.	Java is also an <u>object-oriented</u> language. However, everything (except fundamental types) is an object in Java. It is a single root hierarchy as everything gets derived from java.lang.Object.

Java doesn't support default arguments like C++.

Java does not support header files like C++.

Java uses the import keyword to include different classes and methods



C++ Program Example

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello C++ Programming";
    return 0;
}
```

File: Simple.cpp

Java Program Example

```
class Simple{
    public static void main(String args[]){
        System.out.println("Hello Java");
    }
}
```

File: Simple.java