Java 8 – 24

By

Dinakar

INDEX

12	Java 19	16 - 17
13	Java 20	17 - 18
14	Java 21	18 - 19
15	Java 22	20 - 21
16	Java 23	21 - 22
17	Java 24	22 - 26

Java 8:

Version	Feature Name	Type	Purpose / Use Case
Java 8	Lambda Expressions	Final	Functional-style programming; simplify anonymous classes
	Stream API	Final	Process collections declaratively (map, filter, reduce)
	Default & Static Methods in Interfaces	Final	Add new methods to interfaces without breaking existing code
	Functional Interfaces + `@FunctionalInterface`	Final	Support lambdas with single abstract method interfaces
	Optional	Final	Avoid `null`; represent optional values clearly
	Date & Time API (`java.time`)	Final	Modern, immutable time/date handling
	Nashorn JS Engine	Final (removed later)	Embed JS inside Java apps

	Repeating Annotations	Final	Apply same annotation multiple times
	Method References	Final	Pass methods like lambdas: `ClassName::methodName`
	Collectors	Final	Combine stream results — e.g., `Collectors.toList()`

Java 9:

Version	Feature Name	Type	Purpose / Use Case
Java 9	JShell (REPL)	Final	Interactive tool to quickly run and test Java code snippets
	Module System (Project Jigsaw)	Final	Strong encapsulation, modular JDK, scalable applications
	`varhandles` (Variable Handles)	Final	Low-level concurrency and atomic operations without reflection
	Private Interface Methods	Final	Allow code reuse inside interfaces without exposing to users
	Factory Methods for Collections	Final	Create immutable lists/sets/maps easily (`List.of()`, `Set.of()`, `Map.of()`)

	Stream API Enhancements	Final	Added `takeWhile`, `dropWhile`, `iterate` with predicates
	Optional API Enhancements	Final	New methods like `ifPresentOrElse()`, `or()`, `stream()`
	Process API Updates	Final	Manage and control operating system processes better
	HTTP/2 Client (Incubator)	Incubator → Final in 11	Modern HTTP/2 & WebSocket support
	Multi-Release JARs	Final	Package different class versions for different Java runtimes in one JAR
	Compact Strings	Final	Optimize memory by storing strings in UTF-8/Latin-1 instead of UTF-16 where possible

Java 10:

Version	Feature Name	Type	Purpose / Use Case
Java 10	Local Variable Type Inference (var)	Final	Simplify code by letting the compiler infer local variable types automatically
	Application Class-Data Sharing (AppCDS)	Final	Reduce startup time and memory usage by sharing common class metadata
	Garbage Collector Interface	Final	Standardize GC implementations for easier development and swapping of collectors

	Parallel Full GC for G1	Final	Improve G1 GC performance by parallelizing full GC phases
	Heap Allocation on Alternative Devices	Final	Allow allocating Java object heap on alternate memory devices (e.g., NV-DIMM)
	Thread-Local Handshakes	Final	Improve thread management with stop-the-world safepoints at thread level
	Root Certificates	Final	Include default set of root certification authorities in JDK
	Experimental Java-Based JIT Compiler (Graal)	Experimental	Enable Graal compiler for better optimization and performance

Java 11:

Version	Feature Name	Type	Purpose / Use Case
Java 11	var in Lambda Parameters	Final	Allow `var` in lambda parameters for consistency and annotations
	New `String` Methods	Final	Convenience methods: `isBlank()`, `lines()`, `strip()`, `repeat()`
	New `Files` Methods	Final	Easier file handling with `Files.readString()` and `Files.writeString()`
	`Optional.isEmpty()`	Final	Opposite of `isPresent()`, improves readability

	HTTP Client (Standardized)	Final	Modern, asynchronous HTTP/2 client for REST APIs
	Flight Recorder (JFR)	Final (Open-sourced)	Low-overhead data collection for profiling and monitoring
	Nest-Based Access Control	Final	Simplify access between nested classes without synthetic bridges
	Dynamic Class-File Constants	Final	More efficient constant pool entries, reduce memory footprint
	Deprecations / Removals	Final	Removal of Java EE & CORBA modules; deprecated Nashorn JS engine
	Single-File Source Code Execution	Final	Run Java files directly with `java HelloWorld.java` (no explicit compilation step)

Java 12:

Version	Feature Name	Type	Purpose / Use Case
Java 12	Switch Expressions (Preview)	Language (Preview)	Simplify `switch` with expressions returning values, reduce boilerplate
	Shenandoah GC	Experimental GC	Low-pause-time garbage collector for large heap applications
	Microbenchmark Suite (JMH-based)	Tooling	Built-in microbenchmarking for performance testing

	JVM Constants API	Final	Model nominal descriptors for class-file constants (dynamic constants support)
	Abortable Mixed Collections for G1 GC	Final	Reduce pause times by aborting long G1 GC cycles
	Promptly Return Unused Committed Memory	Final	Improve memory efficiency by returning unused heap memory to OS
	Default CDS Archives	Final	Improve startup time with pre-generated class-data sharing archives

Java 13:

Version	Feature Name	Type	Purpose / Use Case
Java 13	Text Blocks (Preview)	Language (Preview)	Multiline strings with `"""`; easier formatting of JSON, SQL, HTML
	Switch Expressions (2nd Preview)	Language (Preview)	Refined version of switch expressions with `yield` keyword
	Reimplementation of Legacy Socket API	Final	Modernize old blocking socket implementation for maintainability
	Dynamic CDS Archives	Final	Extend CDS by generating class-data archives after application runs

	ZGC: Uncommit Unused Memory	Final	Return unused heap memory to OS when idle

Java 14:

Version	Feature Name	Type	Purpose / Use Case
Java 14	Switch Expressions (Final)	Language (Final)	Finalized switch expressions with concise syntax and `yield` keyword
	Records (Preview)	Language (Preview)	Immutable data carrier classes with minimal boilerplate
	Pattern Matching for `instanceof` (Preview)	Language (Preview)	Simplify `instanceof` checks by eliminating explicit casting
	Helpful NullPointerExceptions	JVM Improvement	More detailed error messages with variable/field name causing NPE
	JFR Event Streaming	Tooling/API	Continuous data streaming from Java Flight Recorder

	Packaging Tool (Incubator)	Tooling (Incubator)	Package self-contained Java applications for OS platforms
	NUMA-Aware Memory Allocation (G1 GC)	JVM/GC Improvement	Optimize memory allocation on NUMA-based systems
	Foreign-Memory Access API (Incubator)	API (Incubator)	Safe, efficient access to native memory outside Java heap

Java 15:

Version	Feature Name	Type	Purpose / Use Case
Java 15	Sealed Classes (Preview)	Language (Preview)	Restrict which classes can extend/implement a class or interface
	Records (Second Preview)	Language (Preview)	Continue refining immutable data carrier classes
	Text Blocks (Final)	Language (Final)	Multi-line string literals with clean indentation and formatting
	Pattern Matching for `instanceof` (Second Preview)	Language (Preview)	Further refine type-checking simplification with `instanceof`
	Hidden Classes	JVM Improvement	Classes not discoverable via reflection, useful for frameworks and dynamic proxies

	Z Garbage Collector (ZGC) Improvements	JVM/GC Improvement	ZGC now production-ready across all platforms, supporting large heaps
	Shenandoah GC (Production)	JVM/GC Improvement	Low-pause-time garbage collector available as a production feature
	Deprecated RMI Activation System	Deprecation	RMI Activation removed for future simplification
	Nashorn JavaScript Engine Removal	Removal	Removed deprecated JS engine introduced in Java 8
	EdDSA Cryptography API	API/Security	Added support for modern Edwards-Curve Digital Signature Algorithm

Java 16:

Version	Feature Name	Type	Purpose / Use Case
Java 16	Records (Final)	Language (Final)	Immutable data classes finalized after previews
	Pattern Matching for `instanceof` (Final)	Language (Final)	Simplifies type casts after `instanceof` checks
	Sealed Classes (Second Preview)	Language (Preview)	Further refinement of controlling class inheritance

	JEP 376: ZGC Concurrent Thread-Stack Processing	JVM/GC Improvement	Improves scalability of ZGC
	JEP 387: Elastic Metaspace	JVM Improvement	Returns unused HotSpot metaspace memory to OS more efficiently
	JEP 338: Vector API (Incubator)	API/Performance	Introduces SIMD (Single Instruction, Multiple Data) vector computations
	JEP 394: Strongly Encapsulate JDK Internals	JVM/Module System	Prevents deep reflection access to JDK internals by default
	JEP 390: Warnings for Value-Based Classes	Language/Warning	Warns when inappropriate operations are used with value-based classes
	JEP 347: C++14 Language Features	JVM Implementation	HotSpot codebase modernized to C++14
	JEP 380: Unix-Domain Socket Channels	API/Networking	Adds Unix-domain socket support for better IPC on Unix-like systems
	JEP 386: Alpine Linux Port	Platform Port	Native port of JDK to Alpine Linux (musl + x64)
	JEP 388: Windows/AArch64 Port	Platform Port	Adds JDK support for Windows on ARM64

Java 17:

Version	Feature Name	Type	Purpose / Use Case

Java 17	Sealed Classes (Final)	Language (Final)	Restricts which classes can extend or implement a superclass/interface
	Pattern Matching for `switch` (Preview)	Language (Preview)	Simplifies switch expressions with type patterns
	JEP 356: Enhanced Pseudo-Random Number Generators	API/Math	Adds new interfaces and algorithms for random number generation
	JEP 382: New macOS Rendering Pipeline	JVM/Graphics	Replaces Quartz pipeline with Metal API for performance
	JEP 389: Foreign Function & Memory API (Incubator)	API/Interop	Access native memory and call native functions without JNI
	JEP 411: Deprecate Security Manager for Removal	Security/Deprecated	Marks Security Manager as deprecated for future removal
	JEP 306: Restore Always-Strict Floating-Point	Language/Math	Floating-point expressions are evaluated strictly according to IEEE 754
	JEP 403: Strongly Encapsulate JDK Internals	JVM/Module System	Prevents access to internal JDK APIs (except some critical ones)
	JEP 406: Pattern Matching for `switch` (Preview)	Language (Preview)	Introduces pattern matching inside `switch` constructs
	JEP 382: Deprecate Applet API for Removal	API/Deprecated	Applet API deprecated; applets officially considered obsolete

	JEP 398: Deprecate RMI Activation for Removal	API/Deprecated	Deprecates outdated RMI Activation system
	JEP 391: macOS/AArch64 Port	Platform Port	Adds JDK support for Apple Silicon (M1/M2 chips)
	JEP 389: Foreign Memory API (Incubator)	API/Interop	Work with off-heap memory safely and efficiently

Java 18:

Version	Feature Name	Type	Purpose / Use Case
Java 18	JEP 400: UTF-8 by Default	JVM/Encoding	Makes UTF-8 the default charset across all platforms for consistency
	JEP 408: Simple Web Server	Tool/Utility	Lightweight web server (for prototyping, testing, education)
	JEP 413: Code Snippets in Java API Documentation	Documentation/Tooling	Allows embedding code snippets in Javadoc with validation

	JEP 416: Reimplement Core Reflection with MethodHandles	JVM/Reflection	Improves performance and maintainability by using `java.lang.invoke` internally
	JEP 417: Vector API (Third Incubator)	API/Performance	Improves vector computations using CPU instructions (SIMD)
	JEP 418: Internet-Address Resolution SPI	API/Networking	Custom service-provider interface for host name/address resolution
	JEP 419: Foreign Function & Memory API (Second Incubator)	API/Interop	Improved API for native memory access and calling native code without JNI
	JEP 420: Pattern Matching for `switch` (Second Preview)	Language (Preview)	Enhances `switch` with type patterns, improves readability
	JEP 421: Deprecate Finalization for Removal	Language/ Memory Mgmt	Marks `finalize()` as deprecated to be removed in future Java

Java 19:

Version	Feature Name	Type	Purpose / Use Case
Java 19	JEP 405: Record Patterns (Preview)	Language (Preview)	Extends pattern matching to deconstruct record values for cleaner data processing

	JEP 422: Linux/RISC-V Port	JVM/Portability	Adds support for Linux on RISC-V hardware architecture
	JEP 424: Foreign Function & Memory API (Preview)	API/Interop	Provides safer and faster access to native code & memory than JNI
	JEP 425: Virtual Threads (Preview)	Concurrency/Performance	Lightweight threads to scale concurrent applications efficiently
	JEP 426: Vector API (Fourth Incubator)	API/Performance	Enhances vector computations with SIMD optimizations
	JEP 427: Pattern Matching for `switch` (Third Preview)	Language (Preview)	Further enhances `switch` with type patterns and exhaustiveness checks
	JEP 428: Structured Concurrency (Incubator)	Concurrency/Programming Model	Simplifies multithreaded programming by handling tasks as a unit

Java 20:

Version	Feature Name	Type	Purpose / Use Case
Java 20	JEP 429: Scoped Values (Incubator)	Concurrency/Memory	Share immutable data across threads safely and efficiently as an alternative to thread-local variables

	JEP 432: Record Patterns (Second Preview)	Language (Preview)	Refines record patterns to destructure record values in more flexible ways
	JEP 433: Pattern Matching for `switch` (Fourth Preview)	Language (Preview)	Adds refinements and stability improvements for pattern-matching in `switch`
	JEP 434: Foreign Function & Memory API (Second Preview)	API/Interop	Updates the API for safer native memory and function access
	JEP 436: Virtual Threads (Second Preview)	Concurrency/Performance	Improves lightweight threads for scalable concurrency
	JEP 437: Structured Concurrency (Second Incubator)	Concurrency/Programming Model	Improves task management by treating multiple threads as a single unit

Java 21:

Version	Feature Name	Type	Purpose / Use Case

Java 21	JEP 430: String Templates (Preview)	Language (Preview)	Allows embedding expressions inside strings safely, improving readability and reducing boilerplate concatenations
	JEP 431: Sequenced Collections	Collections API	Introduces new interfaces to provide consistent iteration order for collections
	JEP 439: Generational ZGC	Garbage Collector	Improves ZGC with generational support for better performance in memory-heavy apps
	JEP 440: Record Patterns	Language Feature	Finalized feature for destructuring record values directly in code
	JEP 441: Pattern Matching for `switch`	Language Feature	Finalized feature enabling type-safe, concise branching using patterns in `switch`
	JEP 442: Foreign Function & Memory API	API/Interop	Finalized API for safe and efficient interop with native code and memory
	JEP 443: Unnamed Patterns and Variables (Preview)	Language (Preview)	Allows ignoring variables with _ to reduce unused variable noise
	JEP 444: Virtual Threads	Concurrency/Performance	Finalized lightweight threads for scalable high-throughput applications (Project Loom milestone)

	JEP 445: Unnamed Classes and Instance Main Methods (Preview)	Language (Preview)	Simplifies writing small programs by removing boilerplate `class` and `main`
	JEP 446: Scoped Values (Preview)	Concurrency/Memory	Safer alternative to thread-local variables for passing immutable data across threads
	JEP 448: Vector API (Sixth Incubator)	Performance/Math API	Provides a way to express vector computations for better CPU performance
	JEP 449: Deprecate Windows 32-bit x86 Port	Platform/Deprecation	Deprecates support for 32-bit x86 Windows builds

Java 22:

Version	Feature Name	Type	Purpose / Use Case
Java 22	JEP 423: Region Pinning for G1	Garbage Collector	Improves G1 GC by reducing latency issues when dealing with critical pinned memory

	JEP 447: Statements before `super(...)` (Preview)	Language (Preview)	Allows code execution before calling superclass constructor → more flexible initialization
	JEP 454: Foreign Function & Memory API	API/Interop (Final)	Finalized API from Project Panama for calling native code and managing memory safely
	JEP 456: Unnamed Variables	Language Feature	Finalized _ as placeholder for unused variables
	JEP 457: Class-File API (Preview)	API (Preview)	Introduces API for parsing, generating, and transforming `.class` files directly
	JEP 458: Launch Multi-File Source-Code Programs (Preview)	Language/Tooling	Enables running programs with multiple `.java` source files without explicit compilation
	JEP 459: String Templates (Second Preview)	Language (Preview)	Improves string templates feature from Java 21
	JEP 460: Vector API (Seventh Incubator)	Performance/Math API	Further refinement of vector computation API for SIMD performance
	JEP 461: Stream Gatherers (Preview)	Streams API (Preview)	Extends Stream API with new `gather()` operator for more flexible stream operations
	JEP 462: Structured Concurrency (Second Preview)	Concurrency (Preview)	Simplifies multithreaded programming by treating tasks as units that run together

	JEP 463: Implicitly Declared Classes & Instance Main (Second Preview)	Language (Preview)	Continues simplifying boilerplate removal for small Java programs
	JEP 464: Scoped Values (Second Preview)	Concurrency /Memory	Iteration on Scoped Values for safe immutable data sharing across threads
	JEP 466: Class-File API (Second Preview)	API (Preview)	Follow-up iteration of new `.class` file API introduced in JEP 457
	JEP 467: Markdown Documentation Comments (Preview)	Language/Docs (Preview)	Allows using **Markdown syntax** inside `/** ... */` Javadoc comments

Java 23:

Version	Feature Name	Type	Purpose / Use Case
Java 23	Stream Gatherers (Preview)	Preview	Extend Stream API with custom aggregation beyond built-in collectors
	Scoped Values (2nd Preview)	Preview	Safer, faster alternative to thread-local variables for sharing data across threads
	Class-File API (Preview)	Preview	Enable parsing, generating, and transforming `.class` files programmatically
	String Templates (3rd Preview)	Preview	Safer, cleaner embedding of variables into strings with processors

	Structured Concurrency (4th Incubator)	Incubator	Manage multiple concurrent tasks as a unit, improving reliability and readability
	Foreign Function & Memory API (7th Incubator)	Incubator	Interact with native code and memory without JNI, safer and faster

Java 24:

Version	Feature Name	Type	Purpose / Use Case
Java 24	Generational Shenandoah (JEP 404)	Experimental (GC)	Explore generational mode for Shenandoah GC to improve performance. (hanno.codes, JRebel, JVM Weekly)
	Compact Object Headers (JEP 450)	Experimental (JVM)	Reduce object header size to shrink memory footprint and enhance locality. (JRebel, Medium)
	Prepare to Restrict Use of JNI (JEP 472)	Deprecation Warning	Warn developers of JNI usage; setting stage for safer native interop. (DEV Community)
	Late Barrier Expansion for G1 (JEP 475)	Performance (GC)	Optimize G1 GC by shifting barrier expansion to improve compilation performance. (hanno.codes, Oracle)

	Key Derivation Function API (JEP 478)	Preview (Security API)	Adds structured API for key derivation, essential for cryptographic applications. (hanno.codes)
	Remove Windows 32-bit x86 Port (JEP 479)	Deprecation / Removal	Drop support for obsolete Windows 32-bit x86 to simplify platform support. (hanno.codes, Medium)
	Ahead-of-Time Class Loading & Linking (JEP 483)	Performance (Runtime)	Cache pre-linked classes to significantly improve startup and warm-up times. (Medium, Oracle)
	Class-File API (JEP 484)	API (Final)	Official API to parse/generate/transform `.class` files — replaces third-party libs. (Medium, JVM Weekly)
	Stream Gatherers (JEP 485)	Streams API (Final)	Boost stream capabilities with stateful intermediate operations like windowing or buffering. (Medium, JVM Weekly)
	Permanently Disable Security Manager (JEP 486)	Removal	Remove legacy Security Manager framework, favoring newer security models. (Medium, bulldogjob.com)
	Scoped Values (JEP 487)	Concurrency (Preview)	Immutable, safe thread-shared values — modern alternative to `ThreadLocal`. (hanno.codes, JVM Weekly)

	Primitive Types in Patterns (JEP 488)	Language (Preview)	Add support for primitive pattern matching in `instanceof` and `switch`. (hanno.codes, pipelinepub.com)
	Vector API (ninth Incubator – JEP 489)	API/Performance	SIMD-enabled numeric operations via vector computations. (JRebel, The Java Version Almanac)
	ZGC: Remove Non-Generational Mode (JEP 490)	GC / Deprecation	Deprecate and remove older ZGC mode in favor of generational ZGC. (DEV Community)
	Synchronize Virtual Threads without Pinning (JEP 491)	Concurrency Fix	Improve virtual thread synchronization by unpinning when blocked in `synchronized` blocks. (Oracle, DEV Community)
	Flexible Constructor Bodies (JEP 492)	Language (Preview)	Permit statements before `super()` or `this()` calls in constructors for flexibility. (JVM Weekly, pipelinepub.com)
	Linking Runtime Images w/o JMODs (JEP 493)	Tooling / Performance	Reduce runtime image size by omitting JMOD files. (Medium, DEV Community)

	Module Import Declarations (JEP 494)	Language (Preview)	Declares module imports for simpler module management. (hanno.codes)
	Simple Source & Instance Main Methods (JEP 495)	Language (Preview)	Allows Java files without explicit classes and `static` main for rapid prototyping. (JVM Weekly, Medium)
	Quantum-Resistant KEM (JEP 496)	Security API	Adds lattice-based key encapsulation — resilient against quantum threats. (hanno.codes, Medium)
	Quantum-Resistant Digital Signature (JEP 497)	Security API	Introduces quantum-safe digital signature algorithm. (hanno.codes, Medium)
	Warn upon Unsafe Memory Access (JEP 498)	Security / Deprecation	Warns developers using unsafe memory methods (`sun.misc.Unsafe`). (DEV Community)
	Structured Concurrency (JEP 499)	Concurrency (Preview)	Manage concurrent tasks as one structured unit to simplify error handling. (JVM Weekly, Medium)

	Deprecate 32-bit x86 Port for Removal (JEP 501)	Deprecation / Removal	Deprecates support for 32-bit x86 architecture across platforms. (JVM Weekly, Wikipedia)