



What is the latest java version ? tell me some features from that.

As of **September 2025**, the latest major release is **Java 25**, which is a **Long-Term Support (LTS)** version.

Java 25 is significant because it finalizes several powerful features that were previously in "preview" mode, focusing on performance, concurrency, and cleaner code.^[1] ^[2]

Key Features of Java 25 (LTS)

1. Performance: Compact Object Headers

- **What it is:** It reduces the size of the metadata header for every Java object from 12 bytes to **8 bytes**.
- **Why it matters:** If your application has millions of small objects (like `String` or `Integer`), this instantly saves massive amounts of RAM (heap memory) without you changing a single line of code.^[3]

2. Clean Code: Flexible Constructor Bodies

- **What it is:** You can now run code *before* calling `super()` in a constructor.
- **Old Way:** You were forced to call `super()` as the very first line, which made validation hard.
- **New Way:**

```
public class Child extends Parent {  
    public Child(int value) {  
        if (value < 0) throw new IllegalArgumentException("Value must be positive");  
        // Validating BEFORE calling the parent!  
        super(value);  
    }  
}
```

This feature is now finalized and ready for production.^[3]

3. Concurrency: Synchronized Virtual Threads

- **What it is:** It solves the "pinning" issue where using the `synchronized` keyword inside a Virtual Thread would block the underlying OS thread, killing performance.
- **Impact:** You can now freely use old synchronized libraries with modern Virtual Threads without worrying about performance bottlenecks.^[4]

4. Project Valhalla: Primitive Types in Patterns

- **What it is:** It enhances `switch` expressions and `instanceof` to handle primitive types (like `int`, `boolean`) directly.
- **Example:**

```
switch (obj) {  
    case int i -> System.out.println("It's an integer: " + i);  
    case boolean b -> System.out.println("It's a boolean: " + b);  
}
```

This removes the need for unnecessary boxing (converting `int` to `Integer`).^[5]

5. Garbage Collection: Generational Shenandoah

- **What it is:** The **Shenandoah GC** (known for ultra-low pause times) now has a "Generational" mode by default.
- **Benefit:** It separates "Young" objects (that die quickly) from "Old" objects. This makes garbage collection much more efficient and faster for standard web applications.^[3]

**

1. <https://www.oracle.com/in/news/announcement/oracle-releases-java-25-2025-09-16/>
2. <https://www.oracle.com/news/announcement/oracle-releases-java-25-2025-09-16/>
3. <https://www.happycoders.eu/java/java-25-features/>
4. <https://www.happycoders.eu/java/java-24-features/>
5. <https://www.infoworld.com/article/3846172/jdk-25-the-new-features-in-java-25.html>
6. <https://www.infoworld.com/article/3491404/jdk-24-the-new-features-in-java-24.html>
7. <https://www.happycoders.eu/java/java-23-features/>
8. <https://symflower.com/en/company/blog/2024/what-is-new-in-java-23/>
9. <https://www.codejava.net/java-se/java-se-versions-history>
10. <https://www.oracle.com/java/technologies/javase/25-relnote-issues.html>
11. <https://www.infoq.com/news/2025/03/java24-released/>
12. https://en.wikipedia.org/wiki/Java_version_history
13. <https://www.java.com/releases/>
14. <https://www.baeldung.com/java-25-features>
15. <https://www.marcobehler.com/guides/a-guide-to-java-versions-and-features>