

Java 24

Must Know Features
If You Don't Want to Be
the “Legacy Coder”



Manav Juneja

Java 24 is here, and if you're still writing Java 8 code in 2024... buddy, it's time for an intervention. 🤪

Let's dive into the coolest features (with code) that will make your life easier—or at least help you flex in tech discussions.

JEP 471

🔥 **1. Computed Constants (JEP 471) - Compile-time evaluated constants! No more runtime calculations clogging your performance.**

Before (Java 23 and Below) :

```
public static final double GRAVITY = 9.81 * 2; // Computed at runtime
```

Now in Java 24

```
public static final computed double GRAVITY = 9.81 * 2; // Evaluated at compile-time 🚀
```

💡 **Performance boost and immutability—what's not to love?**

JEP 463

🥰 2. Implicit Classes & Instance Main Methods (JEP 463) - Java finally lets you skip the public class Main boilerplate.

Before(Java 23 and Below)

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello, Java 23!");  
    }  
}
```

Now in Java 24

```
void main() {  
    System.out.println("Hello, Java 24!");  
}
```

💡 Just type `void main()` and Java handles the rest! Simple, elegant, and less typing.

JEP 456

🛡️ 3. Primitive Types in Generics (JEP 456) - No more unnecessary boxing/unboxing!

Before(Java 23 and Below)

```
List<Integer> numbers = List.of(1, 2, 3); // Boxing slows things down  
int firstNumber = numbers.get(0); // Unboxing happens here
```

Now in Java 24

```
List<int> numbers = List.of(1, 2, 3); // No boxing, direct primitive use 💪
```

💡 Better performance and cleaner code. You love to see it.

JEP 430

🐍 4. String Templates (JEP 430) - No more String.format() nonsense.

Before(Java 23 and Below)

```
String name = "Manav";  
System.out.println("Hello, " + name + "!");
```

Now in Java 24

```
String name = "Manav";  
System.out.println(STR."Hello, \{name}!");
```

💡 **Reads cleaner, looks modern, and finally catches up to JavaScript & Python**

JEP 456

⚡ 5. Unnamed Variables & Patterns (JEP 456) - Skip variables you don't need!

Before(Java 23 and Below)

```
record Person(String name, int age) {}

Person p = new Person("Alice", 25);
if (p instanceof Person var person) {
    System.out.println(person.name());
}
```

Now in Java 24

```
Person p = new Person("Alice", 25);
if (p instanceof Person(_, int age)) { // Ignore 'name'
    System.out.println("Age: " + age);
}
```

💡 **Less clutter, more clarity!**



Manav Juneja

Remaining

 **6. Performance & Garbage Collector Improvements -**
Because Java updates and GC improvements go hand in
hand.

- ◆ **Better memory management in G1 and ZGC.**
- ◆ **Faster warm-up time for Java apps.**
- ◆ **Lower latency means your app runs smoother.**

Follow
For
More



Manav Juneja