
✅ What is a Lambda Expression?

A **lambda expression** is a short block of code that takes parameters and returns a value. It is used primarily to implement **functional interfaces** (interfaces with exactly one abstract method).

📌 Syntax

(parameter1, parameter2) -> { body }

Simplified forms:

Form	Example	When used
No parameters	() -> System.out.println("Hello")	No input
One parameter (no type)	x -> x * x	Type inferred
Multiple params	(a, b) -> a + b	More than one parameter
With types	(int a, int b) -> a + b	Explicit typing
Code block	(x, y) -> { int sum = x + y; return sum; }	Multi-line body

---

📌 Example: Using Lambda with Functional Interface

Java provides the @FunctionalInterface annotation:

```
@FunctionalInterface
interface Operation {
   int apply(int a, int b);
}
```

Using lambda:

```
Operation add = (a, b) -> a + b;
System.out.println(add.apply(5, 3)); // 8
```

📌 Lambda with Java's Built-in Functional Interfaces

1. Predicate — returns boolean

```
Predicate<Integer> isEven = n -> n % 2 == 0;
```

2. Function<T,R> — transforms input to output

```
Function<String, Integer> length = s -> s.length();
```

3. Consumer — takes input, returns nothing

```
Consumer<String> printer = msg -> System.out.println(msg);
```

4. Supplier — no input, returns value

```
Supplier<Double> randomValue = () -> Math.random();
```

📌 Lambda with Collections

forEach

```
List<String> names = List.of("A", "B", "C");

names.forEach(name -> System.out.println(name));
```

sort with Comparator

```
List<Integer> nums = List.of(5, 2, 8, 1);

nums.sort((a, b) -> a - b);
```

Streams

```
List<Integer> nums = List.of(1, 2, 3, 4, 5);


nums.stream()
   .filter(n -> n % 2 == 0)
   .map(n -> n * n)
   .forEach(System.out::println);
```

🎯 Why Use Lambda Expressions?

✓ Reduce boilerplate code
✓ Make code cleaner and more readable
✓ Enable functional programming
✓ Work naturally with Streams API
✓ Useful for callbacks and event handling

```
import java.util.*;

public class LambdaExample {
   public static void main(String[] args) {

      List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5, 6);

      // Lambda expression to print even numbers
      numbers.forEach(n -> {
         if (n % 2 == 0) {
            System.out.println(n + " is even");
         }
      });
   }
}
```