

Java 8 Notes (Simple & Practical)

1. Introduction

Java 8 introduced major features to make code cleaner, faster, and modern.

2. Lambda Expressions

Before Java 8:

```
Runnable r = new Runnable() {  
    public void run() {  
        System.out.println("Hello");  
    }  
};
```

Java 8 (Lambda):

```
Runnable r = () -> System.out.println("Hello");
```

- `() ->` means function without parameters - Shorter & clean

3. Functional Interface

- Interface with **only one abstract method**

```
@FunctionalInterface  
interface Say {  
    void speak();  
}  
Say s = () -> System.out.println("Hi Java 8");  
s.speak();
```

4. Streams API

Example: Filter even numbers from list

```
List<Integer> nums = Arrays.asList(1,2,3,4,5);  
nums.stream()  
    .filter(n -> n % 2 == 0)  
    .forEach(System.out::println); // output: 2 4
```

5. Method Reference

```
nums.forEach(System.out::println); // Shortcut for lambda
```

6. Optional (Avoid NullPointerException)

```
Optional<String> name = Optional.ofNullable(null);  
System.out.println(name.orElse("No Value")); // Output: No Value
```

7. Default & Static Methods in Interface

```
interface Car {  
    default void start() { System.out.println("Car starting..."); }  
    static void info() { System.out.println("This is a Car Interface"); }  
}
```

8. More on Stream API

```
List<String> names = Arrays.asList("Ahmed", "John", "Aman");  
// Convert to uppercase and print  names.stream()  
    .map(String::toUpperCase)  
    .forEach(System.out::println);
```

9. New Date & Time API

```
LocalDate today = LocalDate.now();  
LocalTime time = LocalTime.now();  
LocalDateTime dateTime = LocalDateTime.now();
```

10. Predicate, Consumer, Supplier

```
Predicate<Integer> isAdult = age -> age >= 18;  
Consumer<String> printer = msg -> System.out.println(msg);  
Supplier<Double> random = () -> Math.random();
```

Would you like me to continue with **Optional, Default Methods, Date API, Collectors etc.?**