

Title: Lambdas - Introduction (Java 8)

I. Introduction to Lambdas

A. Definition

1. A lambda expression is a short, concise way to define a function (or method) in Java.
2. Lambdas were introduced in Java 8 to enable functional programming.

B. Key Features

1. Simplify code by reducing verbosity.
2. Allow inline implementation of functional interfaces.
3. Enhance support for parallel processing with Stream API.

II. Syntax

A. Basic Syntax

(parameter_list) -> { method_body }

B. Examples

```
1.(int a, int b) -> {  
    return a + b;  
}  
2.() -> {  
    System.out.println("Hello World!");  
}
```

III. Functional Interfaces

A. Definition

1. An interface with a single abstract method (SAM) is called a functional interface.
2. Lambdas can be used to create instances of functional interfaces.

B. Common Functional Interfaces

1. java.util.function

- a. Predicate
- b. Consumer
- c. Function
- d. Supplier

IV. Usage

- A. Passing Lambda as an argument.
- B. Returning Lambda from a method.
- C. Assigning Lambda to a variable.

V. Method References

A. Used to simplify lambda expressions that call existing methods.

B. Syntax

- ```
1. ClassName::methodName
2. objectReference::methodName
3. super::methodName
4. this::methodName
```

## Quizzes:

### Easy Level:

1. What is the basic syntax of a lambda expression?
  - a. { parameter\_list -> method\_body }
  - b. (parameter\_list) -> { method\_body }
  - c. (parameter\_list -> method\_body)
  - d. [parameter\_list] -> { method\_body }
2. Which Java version introduced lambda expressions?
  - a. Java 6
  - b. Java 7
  - c. Java 8
  - d. Java 9

### Medium Level:

1. What is a functional interface in Java?
  - a. An interface with multiple abstract methods
  - b. An interface with a single abstract method
  - c. An interface with only default methods
  - d. An interface with only static methods
2. Which of the following is NOT a functional interface in the java.util.function package?
  - a. Predicate
  - b. Consumer
  - c. Function
  - d. Comparator

### Tough Level:

1. Given the following code, which lambda expression can replace the commented line?

```
interface MyFunction {
 int apply(int x, int y);
}

// TODO: Replace with lambda
MyFunction myFunction = /* Lambda expression here */;
```

- a. (int x, int y) -> x \* y;
- b. (x, y) -> return x \* y;
- c. (int x, int y) -> { return x \* y; }
- d. (x, y) -> x \* y;

2. Which of the following method reference types refers to an instance method of an existing object?

- a. ClassName::methodName
- b. objectReference::methodName
- c. super::methodName
- d. this::methodName

## Practice Programs:

### Easy Level:

1. Write a program that uses a lambda expression to implement the Runnable interface and print "Hello, Runnable!".

### Medium Level:

1. Write a program that sorts a list of strings in ascending order using a lambda expression with the Collections.sort method.

### Tough Level:

1. Write a program that filters a list of integers based on a given condition using a lambda expression and the java.util.function.Predicate functional interface.