Lambda expressions

Which of the following represents a lambda expression in Java?
A) (int x, int y) -> { return x + y; }
B) int add(int x, int y) { return x + y; }
C) add(int x, int y) -> x + y;
D) int add(x, y) => { return x + y; }

- What is the type of a lambda expression?
- A) int
  B) String
  C) Functional interface
  D) Marker interface

Which of the following functional interfaces represents a lambda expression that takes two integers and returns their sum?

A) BinaryOperator<Integer>

B) UnaryOperator<Integer>

C) Consumer<Integer>

D) Function<Integer, Integer>

Which keyword is used to denote a lambda expression in Java?
A) execute
B) runnable
C) lambda
D) None, it's represented with symbols ->

In a lambda expression (a, b) -> a + b, what does a and b represent?
A) a is the method name and b is the return value.
B) a is the parameter name and b is the method name.
C) a and b are the parameter names.
D) a and b are the method names.

What is the correct syntax of a lambda expression in Java?
A) (parameters) -> expression
B) parameters -> { statements }
C) parameters -> expression
D) All of the above

- Which of the following is true about lambda expressions?
- A) They are executed at runtime
  B) They can be used to replace functional interfaces
  C) They cannot be passed as arguments to methods
  D) They always require explicit type declaration

Which package contains the Predicate interface in Java?
A) java.util.function
B) java.util.lambda
C) java.function
D) java.lambda

- What is the purpose of a functional interface in Java?
- A) To provide an interface for functional programming languages
  B) To define a generic type
  C) To represent a class with only one method
  D) To extend multiple classes

Which of the following represents a valid lambda expression in Java?
A) (x, y) -> { return x + y; }
B) (int x, y) -> { return x + y; }
C) (int x, int y) -> x + y
D) int add(int x, int y) -> { return x + y; }

- How does a lambda expression differ from a regular method?
- A) Lambda expressions cannot have parameters
  B) Lambda expressions cannot have return types
  C) Lambda expressions can be passed as arguments to methods or stored in variables
  D) Lambda expressions cannot be used as arguments for functional interfaces

Which of the following functional interfaces represents a lambda expression that takes an integer and returns void?
A) Function<Integer, Void>
B) Consumer<Integer>
C) Supplier<Void>
D) Predicate<Integer>

What is the correct way to declare a functional interface in Java?

A) interface MyInterface { void myMethod(); }

B) interface MyInterface extends Function { void myMethod(); }

C) interface MyInterface<T> { T myMethod(); }

D) interface MyInterface extends Runnable { void myMethod(); }

Which of the following is true regarding the lambda expression (a, b) -> a + b?

A) It is invalid due to the return statement

B) It can only be used with int data types

C) It represents adding a and b

D) It requires explicit type declaration for a and b

Which method of the Predicate functional interface is used to test a cond

A) apply()

B) check()

C) test()

D) evaluate()