1. What is lambda expression of java 8?

Ans. As it is name suggests it’s an expression which allow you to write more succinct code in java 8. For example (a,b) -> a+b is a lambda expression (look for the arrow -> .

1. Can you pass lambda expressions to a method ? when?

Ans. Yes , you can pass a lambda expression to a method provided it is expecting a functional interface . For example, if a method accepting a Runnable, Comparable or comparator then you can pass a lambda expression to it because all these are functional interfaces in Java 8.

1. What is the functional interface in java 8?

Ans. A functional interface in Java 8 is an interface with a single abstract method. For example, comparator which has just only one abstract method called compare() or Runnable which has just one abstract method called ()run. There are also annotated with @Functionalinterface but that’s optional.

1. Why do we use lambda expressions in Java?

Ans. The main benefit of lambda in Java8 is that now it’s easier to pass a code block to a method.

Earlier, the only way to di this was wrapping the code inside an anonymous class, which requires a lot of boilerplate code.

1. Is it mandatory for a lambda expression to have parameters?

Ans. No, It’s not mandatory for a lambda expression to have parameters, you can define a lambda expression without parameters.