Question :- what is lambda expression in java8?
A lambda expression is a short block of code which takes in parameters and returns a value. Lambda expressions are similar to methods, but they do not need a name and they can be implemented right in the body of a method.

Question :- which method you can pass lambda expression to a method? when?

A lambda expression passed in a method that has an argument of type of functional interface. If we need to pass a lambda expression as an argument, the type of parameter receiving the lambda expression argument must be of a functional interface type.

Question :- what is the functional interface in java 8?

A functional interface is an interface that contains only one abstract method. They can have only one functionality to exhibit. From Java 8 onwards, lambda expressions can be used to represent the instance of a functional interface. A functional interface can have any number of default methods.

Question :- why do we use lambda expression in java?

They provide a clear and concise way to represent one method interface using an expression. Lambda expressions also improve the Collection libraries making it easier to iterate through, filter, and extract data from a Collection . In addition, new concurrency features improve performance in multicore environments.

Question :- is it mandatory for a lambda expression to have parameters?
The lambda expression should have the same number of parameters and the same return type as that method. Java has many of these kinds of interfaces built in, such as the Consumer interface (found in the java.util package) used by lists.