what is lambda expression?

lambda expression is just an anonymous function. that means the function which does not have any name, return type and access modifier. It is used to provide the implementation of functional interface.

Can you pass lambda expression to a method? why?

⇒ Yes, we can pass lambda expression to a method provided it is expecting a function interface. for ex:- if a method is accepting runnable, comparator, comparable then we can pass lambda expression to it. because all of these are functional interface.

What is functional interface?

→ If any interface allows only one abstract method then it is called functional interface. for example:- Comparator which has only one abstract method called compare() or Runnable which has only abstract method called run() and there are many more functional interface introduce in jdk on java.util.function package. They are also annotated with @functional interface but that's optional.

What is the benefit of lambda expression?

→ The main benefit of lambda expression is that its easier to pass a code block to method. Earlier the only way to do this was wrapping code inside an anonymous class which requires a lot of boilerplate code.

Is it manadatory for lambda expression to have parameter?

No, its not manadatory for lambda expression to have parameter we can define a lambda expression without parameters.

Can you write lambda expression directly?

No, we cannot write lambda expression directly because lambda expression is dependent on functional interface. It is used to provide the implementation of functional interface.