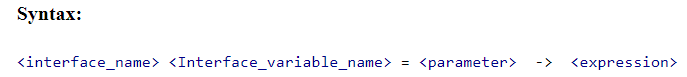
Lambda expressions are code segments that behave like a regular method. They are designed to accept a set of parameters as input and return a value as an output. Unlike methods, lambda expression does not mandatorily require a specific name.



Why Lambda Expression :

Lambda can be created without instantiating a class

Lambda can be treated as an object

It enables functional programming and decrease the number of lines of coding

**Characteristics of lambda expression:**

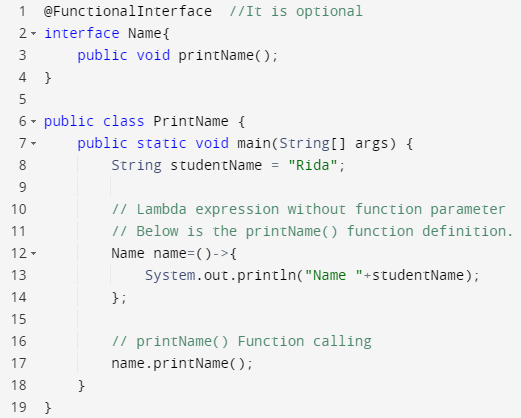
* It is option to declare parameter type declaration while writing lambda expression.
* No need to write circular brackets if we have single parameter.
* No need to write circular brackets if we have single parameter.
* return key word is optional if we don’t have curly braces or it is a single line expression.

**Lambda Expression with no function argument and no return type:**

Below is the example of Lambda expression having no parameters and function return type.

3. How lambda expression and functional interfaces are related?

Functional interface is an interface of java with one abstract method. In addition of creating an interface instance by declaring and instantiating a class, instance of functional interface can be crated with lambda expression.



**OUTPUT:**

Name Rida