LongFunction Functional Interface

LongFunction interface represents a function that accepts long valued argument or operand and produced result of type R. This is long data type (primitive) specialization for Function interface.

We have discussed [Function](http://data-structure-learning.blogspot.com/2015/07/java-lambda-function-functional.html), [BiFunction](http://data-structure-learning.blogspot.com/2015/07/java-lambda-bifunction-functional.html), [DoubleFunction](http://data-structure-learning.blogspot.com/2015/07/java-lambda-doublefunction-functional.html), [DoubleToIntFunction](http://data-structure-learning.blogspot.com/2015/07/java-lambda-doubletointfunction.html), [DoubleToLongFunction](http://data-structure-learning.blogspot.com/2015/07/java-lambda-doubletolongfunction.html) interfaces. I would highly recommend reading them. I have also written on [High Order functions](http://data-structure-learning.blogspot.com/2015/07/higher-order-functions-using-function.html) using Function functional interface.

|  |
| --- |
| **LongFunction Interface Declaration**  **public** **interface** LongFunction<R>  LongFunction interface accepts the object type of R which indicates the result. |
| **apply() method**  R apply(**long** value);  This method is used to apply this function to given argument and return the result. Below example shows us how to use LongFunction interface. It accepts the value as long data type and return the string version of it. Then we take the length of that result.  LongFunction<String> longFunction = (val) -> String.*valueOf*(val);  System.***out***.println(longFunction.apply(10).length()); //Outputs 2 |

That’s all on LongFunction interface.

Read about important java.util.function package’s interface [here](http://data-structure-learning.blogspot.com/p/functional-programming-in-java.html). [Consumer](http://data-structure-learning.blogspot.com/2015/07/java-lambda-consumer-functional.html), [Function](http://data-structure-learning.blogspot.com/2015/07/java-lambda-function-functional.html), [Supplier](http://data-structure-learning.blogspot.com/2015/07/java-lambda-supplier-functional.html), [BinaryOperator](http://data-structure-learning.blogspot.com/2015/07/java-lambda-binaryoperator-functional.html) & [Predicate](http://data-structure-learning.blogspot.com/2015/07/java-lambda-predicate-functional.html) Functional Interfaces. I have also written on [High Order functions](http://data-structure-learning.blogspot.com/2015/07/higher-order-functions-using-function.html) using Function functional interface.