ToDoubleBiFunction Functional Interface

Before we begin to discuss about ToDoubleBiFunction interface we need to know about [BiFunction](http://data-structure-learning.blogspot.com/2015/07/java-lambda-bifunction-functional.html) interface. BiFunction interface represents a function that accepts two arguments *T* and *U* and returns result of type *R*. Input and result are of type Object. Previously we have discussed about BiFunction interface and [Function](http://data-structure-learning.blogspot.com/2015/07/java-lambda-function-functional.html) interface. I would highly recommend you to read it. I also have written on [higher order functions](http://data-structure-learning.blogspot.com/2015/07/higher-order-functions-using-function.html).

ToDoubleBiFunction interface represents a function that accepts two argument of type T and U and returns double-valued result.

|  |
| --- |
| **ToDoubleBiFunction Interface Declaration**  **public** **interface** ToDoubleBiFunction<T, U>  T and U are the input types. |
| **applyAsDouble() method**  **double** applyAsDouble(T t, U u);  T and U are input for the method applyAsDouble(). This function applies the given arguments and returns double result. Let us take simple example of take two float values and using Math.pow method.  ToDoubleBiFunction<Float, Float> aa = (val1, val2) -> Math.*pow*(val1, val2);  **double** res = aa.applyAsDouble(8f, 2f);  System.***out***.println(res); //Outputs 64.0 |

That’s all on ToDoubleBiFunction 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.