Java Lambda Expressions

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.

## Syntax

🡪The simplest lambda expression contains a single parameter and an expression: 

🡪To use more than one parameter, wrap them in parentheses: 

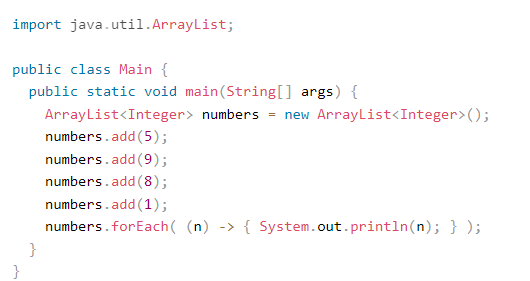
🡪Expressions are limited. They have to immediately return a value, and they cannot contain variables, assignments or statements such as if or for. In order to do more complex operations, a code block can be used with curly braces. If the lambda expression needs to return a value, then the code block should have a return statement.



## Using Lambda Expressions

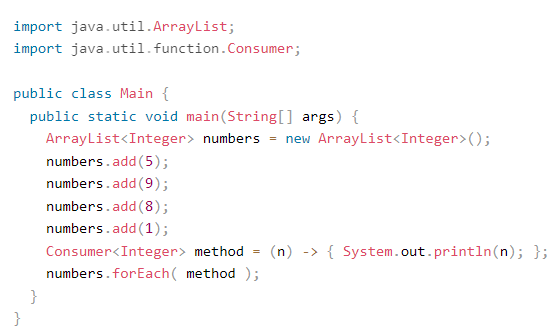
Lambda expressions are usually passed as parameters to a function:

🡺Use a lamba expression in the ArrayList's forEach() method to print every item in the list:



🡪Lambda expressions can be stored in variables if the variable's type is an interface which has only one method. 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.

🡺Use Java's Consumer interface to store a lambda expression in a variable:



🡪To use a lambda expression in a method, the method should have a parameter with a single-method interface as its type. Calling the interface's method will run the lambda expression:

🡺Create a method which takes a lambda expression as a parameter:

