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.

Example

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

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

ArrayList<Integer> numbers = new ArrayList<Integer>();

numbers.add(5);

numbers.add(9);

numbers.add(8);

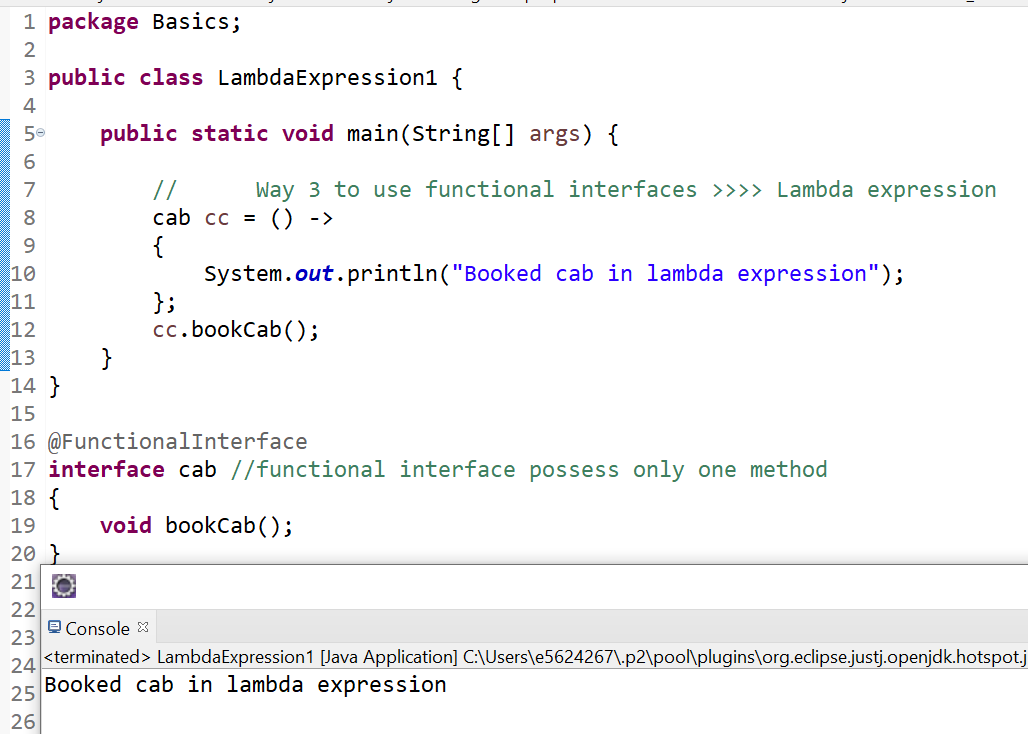
numbers.add(1);

numbers.forEach( (n) -> { System.out.println(n); } );

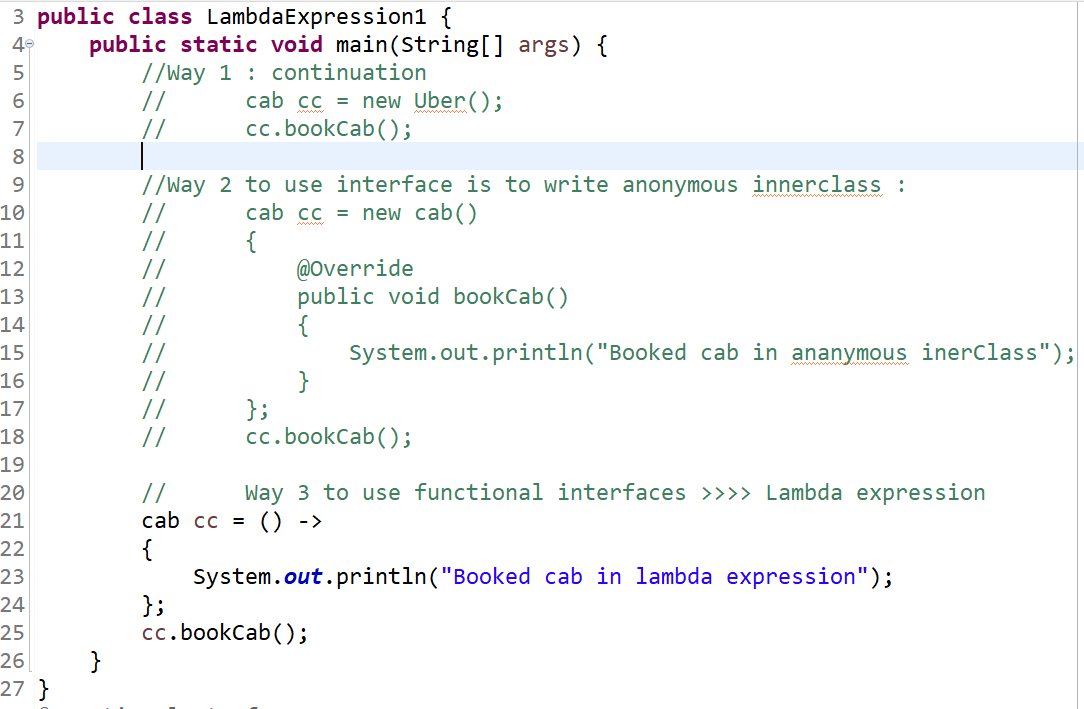
}

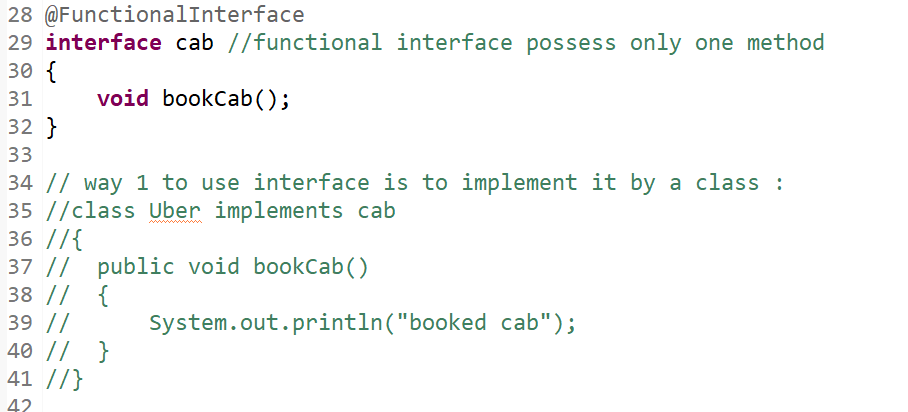
}

Functional interfaces : Interface that contains one single method. Lambda expression only works with functional interfaces. Lambda ex. Can have 0, 1, and multiple parameters.



Other than lambda expression below commented part shows other two ways to access functional interfaces :

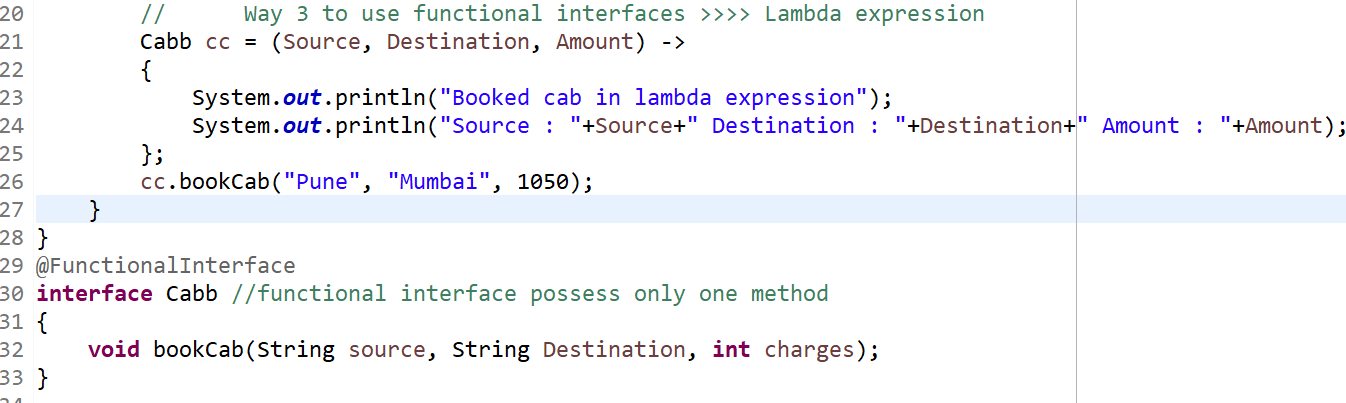




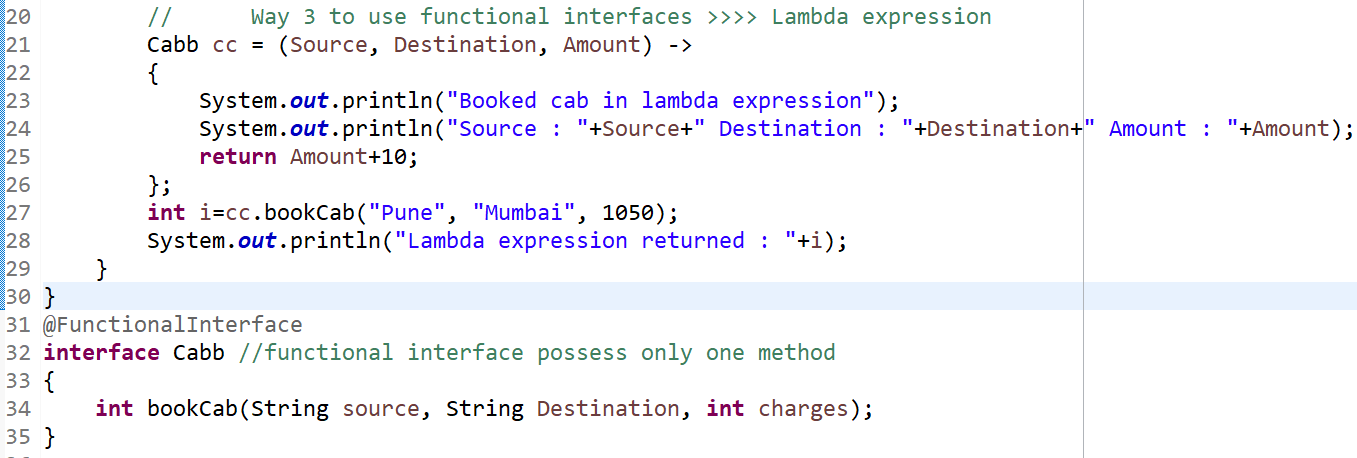
Lambda expression can have :

1. Zero parameter
2. 1 parameter
3. Multiple parameters
4. Can have return type

Example of lambda expression with Multiple parameters :



Example of lambda expression with return type :



Lambda expressions can :

1. Use local variables
2. Static variables
3. Instance variables