JAVA - J2EE Batch 2

Name – Aman Yadav

E-mail : prakashaman5@gmail.com

Phone: +919519131321

# Assignment-6

## 2 PROBLEM STATEMENT – FIND TOTAL OF A GROCERY CART INCLUDING TAXES USING LAMBDA EXPRESSION

**Given a Map and Double find total purchase cost including taxes by iterating through the Map using lambda expression**

**This exercise contains a class named CartCheckout with the following method:**

```
 +billGenerator(Map<String, BigDecimal>, Double) : String
   - Should take a Map and Double as input and return a String as result
   - Should validate the input Map by checking it is empty and return error
String if it is empty
   - Should return error String when Map contains null or empty or blank
space as a value
  - Should return error String when the Double is negative or null
```

### 2.1 EXAMPLE

```
Sample Input:
{Apple=54, Grapes=36.78, Papaya=27.89, Orange=23.6, Banana=10.2}, 10.5

Expected Output:
"168.47935"
```

---

```
Sample Input:
{}, 13

Expected Output:
"The cart Map is empty"
```

---

```
Sample Input:
{Apple=54, Grapes=36.78, Papaya=27.89, Orange=23.6, Banana=10.2}, -2.5

Expected Output:"The taxPercent cannot be negative"
```

---

```
Sample Input:
{Apple=54, Grapes=36.78, Papaya=27.89, Orange=23.6, Banana=10.2}, null

Expected Output:
"The taxPercent cannot be null"
```

→

cartCheckOut.java

```java
package com.main;

import java.math.BigDecimal;
import java.util.Map;

public class cartCheckOut {

    public String billGenerator(Map<String, BigDecimal> items, Double taxPercent) {

        if (items.isEmpty()) {
            return "The cart Map is empty";
        }

        if (taxPercent == null) {
            return "The taxPercent cannot be null";
        } else if (taxPercent < 0) {
            return "The taxPercent cannot be negative";
        }

        BigDecimal totalCost = items.entrySet().stream()
                .filter(entry -> entry.getValue() != null && entry.getValue().compareTo(BigDecimal.ZERO) > 0)
                .map(entry -> entry.getValue()).reduce(BigDecimal.ZERO, BigDecimal::add);

        BigDecimal taxAmount = totalCost.multiply(BigDecimal.valueOf(taxPercent / 100.0));

        BigDecimal totalCostWithTax = totalCost.add(taxAmount);
        return totalCostWithTax.toString();
    }

    public static void main(String[] args) {
        cartCheckOut checkout = new cartCheckOut();

        Map<String, BigDecimal> items1 = Map.of("Apple", BigDecimal.valueOf(54), "Grapes", BigDecimal.valueOf(36.78),
                "Papaya", BigDecimal.valueOf(27.89), "Orange", BigDecimal.valueOf(23.6), "Banana",
                BigDecimal.valueOf(10.2));
        System.out.println(checkout.billGenerator(items1, 10.5));
        Map<String, BigDecimal> items2 = Map.of();
        System.out.println(checkout.billGenerator(items2, 13.0));

        Map<String, BigDecimal> items3 = Map.of("Apple", BigDecimal.valueOf(54), "Grapes", BigDecimal.valueOf(36.78),
                "Papaya", BigDecimal.valueOf(27.89), "Orange", BigDecimal.valueOf(23.6), "Banana",
                BigDecimal.valueOf(10.2));
        System.out.println(checkout.billGenerator(items3, -2.5));
        Map<String, BigDecimal> items4 = Map.of("Apple", BigDecimal.valueOf(54), "Grapes", BigDecimal.valueOf(36.78),
                "Papaya", BigDecimal.valueOf(27.89), "Orange", BigDecimal.valueOf(23.6), "Banana",
                BigDecimal.valueOf(10.2));
        System.out.println(checkout.billGenerator(items4, null));
    }

}
```

Console

```
<terminated> cartCheckOut [Java Application] D:\my\eclipse-jee-2024-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v202
168.47935
The cart Map is empty
The taxPercent cannot be negative
The taxPercent cannot be null
```