Following are the bug fixes

|  |  |  |
| --- | --- | --- |
|  | Bugs | Fix |
|  | No implementation class for List Interface | Added implementation class Arraylist for List Interface |
|  | Constructor for OrderLine does not have reference to current object within an instance | Added keyword this to item and quantity of OrderLine constructor to refer to current class instance variable |
|  | Code for “imported” items is case sensitive | Added toLowercase() method- converts the String to lower case characters |
|  | Imported is misspelled in Order 3 | Corrected to imported |
|  | Quantity for each Orderline is not shown. | Fixed to print quantity for each Orderline as per the output |
|  | ArrayIndexOutofBound Exception for iterating the items in the order. Array has been accessed with an illegal index | Changed to access available index in the array |
|  | Code implements HashMap in which insertion order is not maintained | Implemented LinkedHashMap to maintain insertion order, required as per the given output |
|  | Code uses “c.clear()” function to reuse cart for next order. But the previous order is overwritten with the next new order, as all the orders refer to same memory location of “c”. Therefore only the latest order is persisted.  Calculate is invoked after fetching all the orders in the map | Calculate method is invoked after putting each order in the map. We clear the previous order in the map as soon as we enter a new record in the map. This is because we want to calculate sales tax and total amount for that particular order only and also to maintain given key-value mapping . |
|  | The Sum of orders should be calculated after fetching all the records | grandTotal is returned by calculate() method and Sum of Orders is calculated in the main method. |
|  | Method rounding() does not return closest value, which does not give the correct calculated tax value. | Used Math.round() method to return the closest value, which gives the desired output. |
|  | Math.floor() returns double value which is already handled by rounding() function. | Rounding function returns a double value so no need to use Math.floor(). |
|  | The output indicates Total is excluding the Sales Tax but the statement total = total + totalTax; violets this. | Removed total = total + totalTax; to get Total excluding the Sales tax. |
|  | Sum of orders is calculated for each order, but we need to calculate sum only once after each order has been processed | Calculated Sum of Orders after processing all the given orders. |