|  |  |  |  |
| --- | --- | --- | --- |
| SL | Bug details | Fix | Test Result |
| 1 | In Order class orderliness map was throwing “NullPointerException”. | Map was not initialized. So it was null and we are trying to operate on null object. | passed |
| 2. | In OrderLine class constructor item and quantity member variables are not set properly. | Added  this.item = item  this.quantity = quantity  to fix it. | passed |
| 3. | Reuse cart is not working. Getting same item information in all the order. | In Foo main class after calling clear() the orderliness from the order is getting removed and order is taking only last added orderliness and copying to others. So created new order object. | passed |
| 4 | In calculator class the rounding() is not considering 3rd digit after decimal point whether it is more or less than 5. | Added Math.round() to round it properly. | passed |
| 5. | Iterate throw items in order is throwing “arrayIndexOutOfBound” exception. | ArrayList starts from 0 and maximum index is size()-1. But here we are doing i<=r.size(). It will be i<r.size(). | passed |
| 6. | Calculate totalprice was not giving result. Only showing integer not 2 digit after decimal point. | Math.floor() will Returns the largest integer that is less than or equal to the argument.So remoded Math.round() and added rounding() function. | passed |
| 7. | Calculate total was not giving result. Only showing integer not 2 digit after decimal point. | Math.floor() will Returns the largest integer that is less than or equal to the argument.So remoded Math.round() and added rounding() function. | passed |
| 8. | Calculate grandtotal was not giving result. Only showing integer not 2 digit after decimal point. | Math.floor() will Returns the largest integer that is less than or equal to the argument.So remoded Math.round() and added rounding() function. | passed |
| 9 | Order 3 was not showing proper expected result. | “imported” spelling was wrong. It was “imprtd”. So .contain() function could not find “imported” word and not adding 15% discount. | passed |
| 10 | Order 3 was not showing proper expected result. | .contain() method looks for “imported” keyword and it is case sensitive. But it was “Imported” in the order. So I changed to “imported” to match the expected result. We can make it case insensitive too. | passed |

Besides I think there are some bugs in the results too.

This is expected result.

\*\*\*\*\*\*\*Order 1\*\*\*\*\*\*\*

1 book: 13.74

1 music CD: 16.49

1 chocolate bar: 0.94

Sales Tax: 2.84

Total: 28.33

\*\*\*\*\*\*\*Order 2\*\*\*\*\*\*\*

1 imported box of chocolate: 11.5

1 imported bottle of perfume: 54.62

Sales Tax: 8.62

Total: 57.5

\*\*\*\*\*\*\*Order 3\*\*\*\*\*\*\*

1 Imported bottle of perfume: 32.19

1 bottle of perfume: 20.89

1 packet of headache pills: 10.73

1 box of imported chocolates: 12.94

Sales Tax: 8.77

Total: 67.98

Sum of orders: 153.81

But if you look into the order 1 , order 2 and order 3 total there is some calculation error. So grandtotal is also different from what I got.

I could not solve the clear() method. It was clearing all the previous orderliness. But after pushing into Map it is taking the next value and we are getting redundancy. So I commented that method and created new object of Order each time. I know this is a bad coding.

Testing:

I have tested all the method through Junit. But calculate() method I could not test it. It does not return any value. So Assert will not work. I have to test tax, total tax, total and grand total. I can do it by changing the return type from void to array and after testing I can write it back. Through Mockito it will not be tested very well as mockito will cerate a mock object and we will set the behavior. But we will never get the values.