

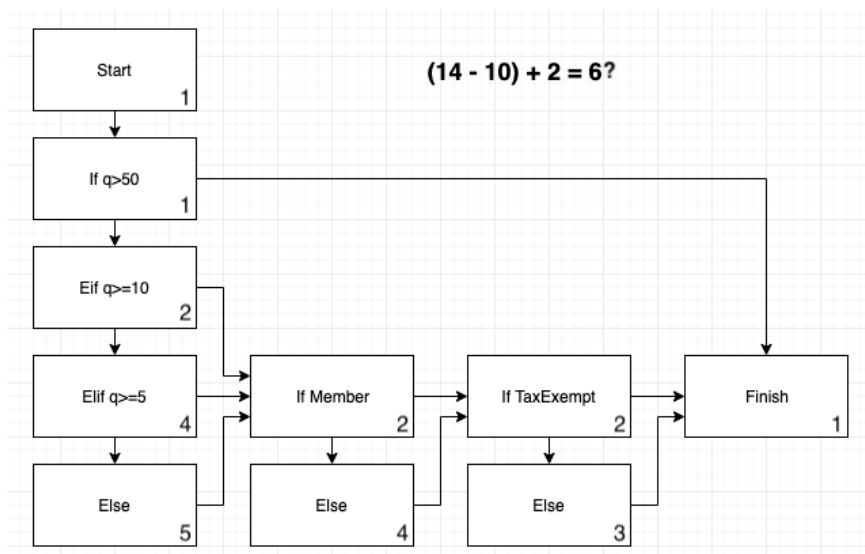
Jess Conway
13 February 2019
COMP 474 – Stone
Homework 1.1 – Software Testing, Part 1

a)

For full coverage, we need tests covering the following parameters:

1. Prices p such that $p > 0$
2. Quantities q such that $q > 50$, $50 \geq q \geq 10$, $10 > q \geq 5$, and $5 > q$
3. DiscountClub values dc such that $dc = \text{true}$, $dc = \text{false}$
4. TaxExempt values te such that $te = \text{true}$, $te = \text{false}$

If we wanted to do this the most efficient way possible, we would only need 6 tests (see flow graph for calculation). However, in practice it appears that 5 tests may be sufficient (the number in each box below corresponds to the number of the test that covers that path). This indicates an error in the graph or missed coverage in the tests below.



Define the checkout method as follows:

- IN: checkout(double price, int quantity, boolean discountMember, boolean taxExempt)
- OUT: netPurchase, netDiscount, taxAmount, netTotal

Sufficient tests for part (a) would be:

Test Number	Test Definition	Expected Output
1	checkout(83.10, 51, true, true)	TooManyItemsError
2	checkout(83.10, 35, true, true)	83.10, 15.79, 0, 67.31
3	checkout(83.10, 35, true, false)	83.10, 15.79, 3.03, 70.34
4	checkout(83.10, 7, false, true)	83.10, 4.16, 0, 78.94
5	checkout(83.10, 3, false, false)	83.10, 0, 0, 86.84

b)

We should also consider the following parameters for boundary analysis:

1. Prices $p=0$, $p=0.005$
 - a. If necessary, we could test only $p=0$
2. Quantities $q=51$, $q=50$, $q=49$, $q=11$, $q=10$, $q=9$, $q=6$, $q=5$, $q=4$, $q=1$, $q=0$
 - a. If necessary, we could test only $q=51$, $q=50$, $q=10$, $q=5$, and $q=0$

Sufficient tests for part (b) are as follows:

Test Number	Test Definition	Expected Output
6	checkout(0, 25, true, true)	NoPriceError
7	checkout(0.005, 25, false, true)	0.01, 0, 0, 0.01
8	checkout(83.10, 51, false, true)	TooManyItemsError
9	checkout(83.10, 50, false, true)	83.10, 8.31, 0, 74.79
10	checkout(83.10, 11, false, true)	83.10, 8.31, 0, 74.79
11	checkout(83.10, 10, false, true)	83.10, 8.31, 0, 74.79
12	checkout(83.10, 9, false, true)	83.10, 4.15, 0, 78.95
13	checkout(83.10, 6, false, true)	83.10, 4.15, 0, 78.95
14	checkout(83.10, 5, false, true)	83.10, 4.15, 0, 78.95
15	checkout(83.10, 4, false, true)	83.10, 0, 0, 83.10
16	checkout(83.10, 1, false, true)	83.10, 0, 0, 83.10
17	checkout(83.10, 0, false, true)	EmptyCartError

*Note that the quantity parameter could be changed for tests 1-2. In addition, the price parameter could be changed any of the tests 8-17, and the membership or tax parameters could be changed for any of these tests. This is because tests 1-2 test boundary values for price, while the rest test boundary values for quantity. None of these tests are concerned about membership or tax status.

c)

In order to check if the rounding method works correctly, we need to check the following:

1. Check that the outputs have exactly two decimals
2. Check that the decimal value is correctly rounded

In addition to this, we should check the following things during BVA and to examine edge cases:

1. Check values where third decimal of input is "5"
2. Check values where third decimal of input is "0"
3. Check values where input has less than 3 decimal places

Define the roundOff method as follows:

- IN: roundOff(double rounded)
- OUT: roundedNumberToTwoDecimals

Sufficient tests to cover part (c) include:

Test Number	Test Definition	Expected Output
18	roundOff(123.4567)	123.46
19	roundOff(98765.4321)	98765.43
20	roundOff(555.5555)	555.56
21	roundOff(10.000)	10.00
22	roundOff(5.52)	5.52
23	roundOff(9.4)	9.40
24	roundOff(7)	7.00

**See that tests 18-19 are general tests for coverage. Meanwhile, tests 20-21 check boundary cases and tests 22-24 test edge cases.