

Team Name: QUN

Members: Kong Jie Wei, Joash; Zhuoyun Xu; Goh Ying Ting; Quentin Le Bars

Account creation - statement coverage testing

(1) Source listing

The section of code to be tested for account creation is shown below. This code is obtained from the TransactionProcessor class.

```
55         // creates new account in list of accData
56         private static void create(String accNum, String accName) {
57             accData.put(accNum, new Account(accNum, "0", accName));
58         }
```

(2) Analysis of test case

Statement coverage testing is used for testing create function in TransactionProcessor class. This white box testing method ensures that every statement in the program is executed at least once. As such, a test case must be made for each statement in the program until every statement has been tested.

Test Case Name	Statements Executed	Account number	Account name
createAccTest1	56, 57, 58	1234567	qwerty

As the first test case is sufficient to cover all the statements, we only have 1 test case.

(3) Listing of actual test inputs

The following test inputs were used for the test case.

Test Case Name	Old Master Accounts File	Transaction Summary File
createAccTest1	1234567 200000 qwerty	NEW 1234568 000 0000000 qwertyu EOS 0000000 000 0000000 ***

(4) Test Report

The following table shows the results of the test case.

Test Case Name	What it was testing	How the output was wrong	What was the error
createAccTest1	Whether a new account could be created successfully in the backend	The output amount of the old master accounts file for the newly created account was 0 instead of 000.	The file processor did not perform string formatting for amounts less than 100. The file processor was modified to take this into account. The subsequent test proved to be successful.

Team Name: QUN

Members: Kong Jie Wei, Joash; Zhuoyun Xu; Goh Ying Ting; Quentin Le Bars

Withdraw - Decision coverage testing

(1) Source listing

The section of code to be tested for withdrawal is shown below. This code is obtained from the TransactionProcessor class.

```
72         // withdraws money from relevant account
73         private static void withdraw(String accNum, int amount) {
74             Account acc = accData.get(accNum);
75             int newBalance = acc.getAmount() - amount;
76             if (newBalance >= 0) {
77                 acc.setAmount(newBalance);
78             } else {
79                 crash("Transaction ignored. " + accNum + " would have a negative balance.");
80             }
81         }
```

(2) Analysis of test case

Decision coverage testing is used for testing the withdraw function in TransactionProcessor class. This white box testing method causes every decision in the program to be made both ways. As such, a test case has to be designed to ensure that each decision is exercised in the program each way, until there is a test case for each side of the decision.

As there is one decision point in line 76, two test cases are generated so that the decision point will be exercised both ways.

Test Case Name	Decision Point	Amount in account	Amount to be withdrawn
withdrawTest1	Line 76, True	200000	100000
withdrawTest2	Line 76, False	100000	100001

(3) Listing of actual test inputs

The following test inputs were used for the test case.

Test Case Name	Old Master Accounts File	Transaction Summary File
withdrawTest1	1234567 200000 qwerty	WDR 1234567 100000 0000000 *** EOS 0000000 000 0000000 ***

Team Name: QUN

Members: Kong Jie Wei, Joash; Zhuoyun Xu; Goh Ying Ting; Quentin Le Bars

withdrawTest2	1234567 100000 qwerty	WDR 1234567 100001 0000000 *** EOS 0000000 000 0000000 ***
---------------	-----------------------	---

(4) Test Report

Test Case Name	What it was testing	How the output was wrong	What was the error
withdrawTest1	Withdrawal is valid if the amount in the account is greater than the amount to be withdrawn	The Valid Account List file could not be found.	The directory where the Valid Account List file was produced was different from where we expected it to be. We modified the directory of the Valid Account List file to resolve this problem.
withdrawTest1	Withdrawal is valid if the amount in the account is greater than the amount to be withdrawn	Valid Account List file was missing '0000000'.	The file processor did not write out the last '0000000' line in the Valid Account List file. The file processor was modified to take this into account. The subsequent test proved to be successful.
withdrawTest2	Withdrawal is invalid if the amount in the account is less than the amount to be withdrawn	The system exit function prevented the maven test function from executing.	The program terminated unexpectedly, which caused the maven program to be unable to continue running. To resolve this, we modified our crash function such that when an invalid input was encountered, the program output the original master accounts file and the corresponding valid account list file, without using system exit. The subsequent test proved to be successful.