A5 – Back Office White Box Testing

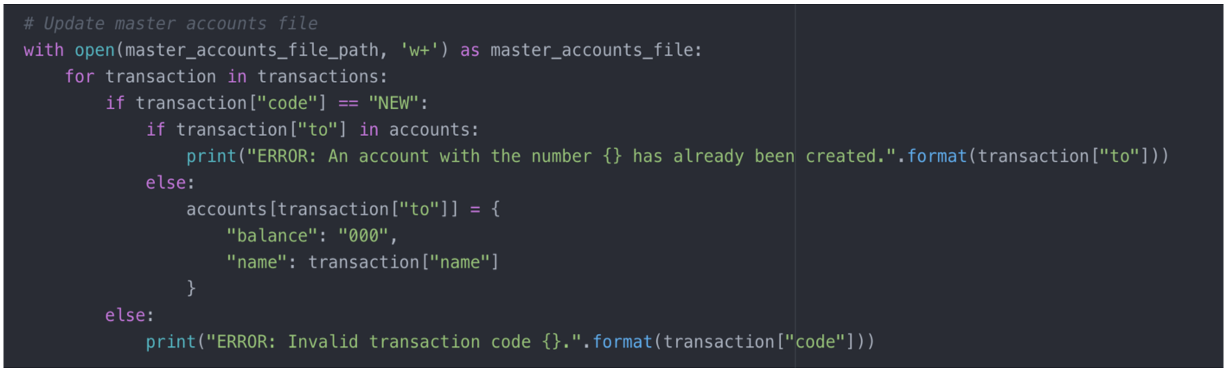
Group 4: All Toppings

Table of Contents

Report 1 – Basic Block Coverage Testing for Account Creation Transactions

Source Listing

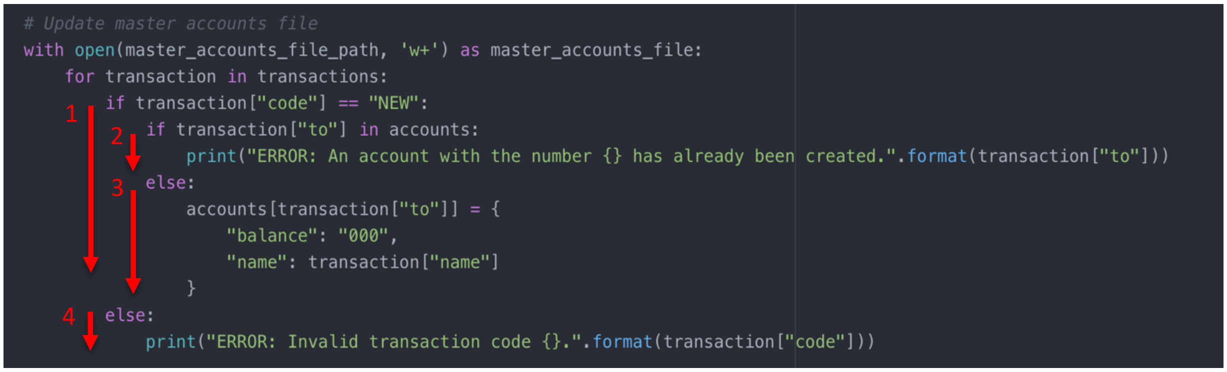
Code section for account creation transactions within *main()* (from *qa327/backoffiice.py*)



```
# Update master accounts file
with open(master_accounts_file_path, 'w+') as master_accounts_file:
    for transaction in transactions:
        if transaction["code"] == "NEW":
            if transaction["to"] in accounts:
                print("ERROR: An account with the number {} has already been created.".format(transaction["to"]))
            else:
                accounts[transaction["to"]] = {
                    "balance": "000",
                    "name": transaction["name"]
                }
        else:
            print("ERROR: Invalid transaction code {}.".format(transaction["code"]))
```

Analysis of Test Cases

The white box testing method chosen is **basic block coverage testing**. The source listing is dissected into 5 sub-blocks as follows:



```
# Update master accounts file
with open(master_accounts_file_path, 'w+') as master_accounts_file:
    for transaction in transactions:
        if transaction["code"] == "NEW":
            if transaction["to"] in accounts:
                print("ERROR: An account with the number {} has already been created.".format(transaction["to"]))
            else:
                accounts[transaction["to"]] = {
                    "balance": "000",
                    "name": transaction["name"]
                }
        else:
            print("ERROR: Invalid transaction code {}.".format(transaction["code"]))
```

The following table shows how each block will be tested to ensure proper code coverage:

Block	Old Master Accounts File	transaction[“code”]	transaction[“to”]	transaction[“name”]	Test	New Master Accounts File
1	8953935 000 Rob	NEW	4093853	Tom	T1	8953935 000 Rob 4093853 000 Tom
2	8953935 000 Rob	NEW	8953935	Rob	T2	“ERROR: An account with the number 8953935 has already been created.”
3	4983952 000 Chad	NEW	8953935	Rob	T3	8953935 000 Rob 4983952 000 Chad
4	4983952 000 Chad	JRM	4983952	Chad	T4	“ERROR: Invalid transaction code JRM”

Actual Test Inputs

Test #	Transaction Line	Output: Master Accounts File	Output: Console
T1	NEW 4093853 000 0000000 Tom	8953935 000 Rob 4093853 000 Tom	N/A
T2	NEW 8953935 000 0000000 Rob	8953935 000 Rob	“ERROR: An account with the number 8953935 has already been created.”
T3	NEW 8953935 000 0000000 Rob	8953935 000 Rob 4983952 000 Chad	N/A
T4	JRM 4983952 000 0000000 Chad	4983952 000 Chad	“ERROR: Invalid transaction code JRM”

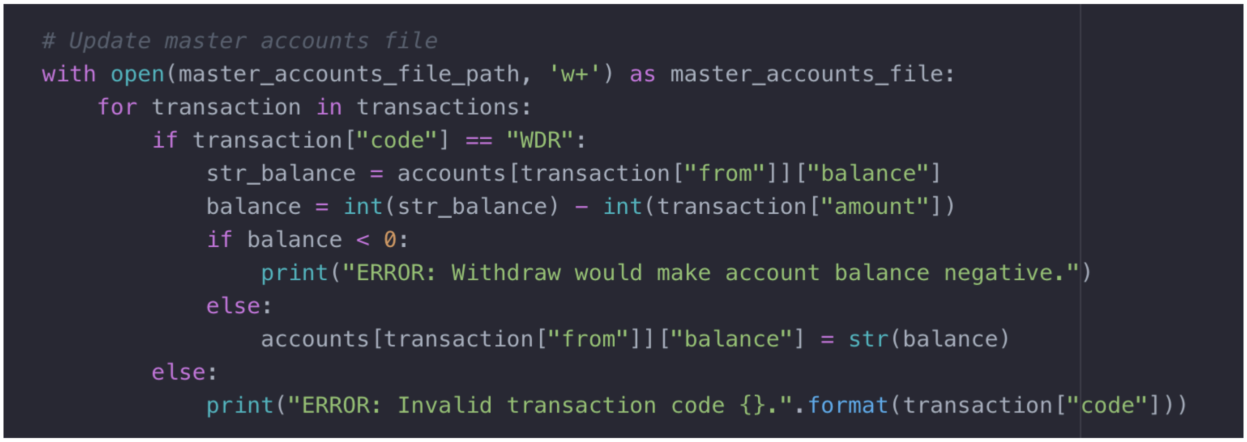
Test Report

Test #	Result: Master Accounts File	Result: Console	Outcome (Pass or Fail)
T1	8953935 000 Rob 4093853 000 Tom		PASS
T2	8953935 000 Rob	ERROR: An account with the number 8953935 has already been created.	PASS
T3	8953935 000 Rob 4983952 000 Chad		PASS
T4	4983952 000 Chad	ERROR: Invalid transaction code JRM	PASS

Report 2 – Decision Coverage Testing for Withdraw Transactions

Source Listing

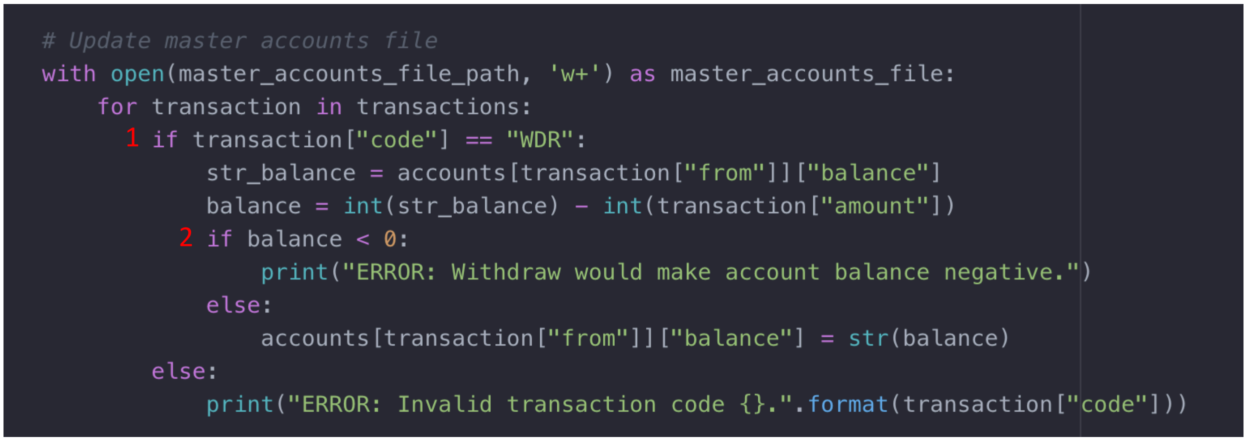
Code section for all withdraw transactions within *main()* (from *qa327/backoffice.py)*



```
# Update master accounts file
with open(master_accounts_file_path, 'w+') as master_accounts_file:
    for transaction in transactions:
        if transaction["code"] == "WDR":
            str_balance = accounts[transaction["from"]]["balance"]
            balance = int(str_balance) - int(transaction["amount"])
            if balance < 0:
                print("ERROR: Withdraw would make account balance negative.")
            else:
                accounts[transaction["from"]]["balance"] = str(balance)
        else:
            print("ERROR: Invalid transaction code {}.".format(transaction["code"]))
```

Analysis of Test Cases

The white box testing method chosen is **decision coverage testing**. The source listing is dissected into 2 decisions:



```
# Update master accounts file
with open(master_accounts_file_path, 'w+') as master_accounts_file:
    for transaction in transactions:
      1 if transaction["code"] == "WDR":
            str_balance = accounts[transaction["from"]]["balance"]
            balance = int(str_balance) - int(transaction["amount"])
          2 if balance < 0:
                print("ERROR: Withdraw would make account balance negative.")
            else:
                accounts[transaction["from"]]["balance"] = str(balance)
        else:
            print("ERROR: Invalid transaction code {}.".format(transaction["code"]))
```

The following table shows how each decision will be tested to ensure proper code coverage:

Decision	**Old Master Accounts File**	**transaction["code"]**	**balance**	**Test**	**New Master Accounts File**
1: true	8953935 493 Rob	WDR	493	T1	8953935 493 Rob
1: false	8953935 493 Rob	KJF	493	T2	"ERROR: Invalid transaction code KJF."
2: true	4983952 000 Chad	WDR	-490	T3	"ERROR: Withdraw would make account balance negative."
2: false	4983952 490 Chad	WDR	100	T4	4983952 390 Chad

Actual Test Inputs

Test #	**Transaction Line**	**Output: Master Accounts File**	**Output: Console**
T1	WDR 0000000 000 8953935 Rob	8953935 493 Rob	N/A
T2	KJF 0000000 000 8953935 Rob	8953935 493 Rob	"ERROR: Invalid transaction code KJF."
T3	WDR 0000000 000 4983952 Chad	4983952 000 Chad	"ERROR: Withdraw would make account balance negative."
T4	WDR 0000000 100 4983952 Chad	4983952 390 Chad	N/A

Test Report

Test #	**Result: Master Accounts File**	**Result: Console**	**Outcome (Pass or Fail)**
T1	8953935 493 Rob		PASS

T2	8953935 493 Rob	ERROR: Invalid transaction code KJF.	PASS
T3	4983952 000 Chad	ERROR: Withdraw would make account balance negative.	PASS
T4	4983952 390 Chad		PASS