Assignment 5 - White Box Testing
Decision Coverage Method for Withdraw

1) Code Sections (in *Backend.py*)

```
# function input transaction summary file and master account file
# if transaction summary valid then return valid master account file data (data type: dictionary)
# if transaction summary not valid then immediately stop
def get_new_master_accounts(self, file1, file2):
    master_list = self.read_master_accounts(file1)
    trans_list = self.read_transaction(file2)
    for i in trans_list:
        if i[0] == "DEP":
            if i[1] in master_list:
                balance = int(master_list[i[1]][0]) + int(i[2])
                master_list[i[1]][0] = str(balance)
            else:
                print("Error! " + str(i[1]) + " not in the master account file!")
                return False

    elif i[0] == "WDR":
        if i[1] in master_list:
            balance = int(master_list[i[1]][0]) - int(i[2])
            if balance < 0:
                print("Error! " + str(i[1]) + " have a negative balance!")
                return False
            else:
                master_list[i[1]][0] = str(balance)
        else:
            print("Error! " + str(i[1]) + " not in the master account file!")
            return False


    for i in master_list:
        line = str(i) + " " + master_list[i][0] + " " + master_list[i][1] + "\n"
    if len(line) > 47:  # check each line have more than 47 characters or not
        print("Error! Each line is at most 47 characters!")
        return False

    return master_list  # return a valid master dictionary
```

2.1) Analysis of Test Cases

In the code for *withdraw*, there are branches (if statements) to complement the constraints, such as non-negative balance, valid account number and character limit for each line in *MasterAccountsFile.txt*. Decision coverage method covers both sides (true and false) of each decision, which is appropriate to test the constraints and requirements. The completion criterion is to design a test case for each side of each decision. In *Backend.py*, there are 4 decisions for withdraw section.

2.2) Design of the Test Cases:

For all cases, the input files are *TransactionSummaryFile.txt* and *MasterAccountsFile.txt*.

Notes for the table:
-i in trans_list: the element in *trans_list*, which is the input reading from *TransactionSummaryFile.txt*
e.g. ['WDR', '1234567', '100000', '0000000', 'theo']
-i[0]: the transaction code, e.g. “NEW”
-i[1]: the account number, e.g. “1234567”
-i[2]: the withdraw amount, e.g. “200000”

Decision	i[0] input	i[1] input	i[2] input	Test	i[0]	i[1]	i[2]
1: true	“WDR”	account number	withdraw amount	T1	“WDR”	account number	withdraw amount
1: false	transaction code other than “WDR”	account number	withdraw amount	T2	transaction code other than “WDR”	account number	withdraw amount
2: true	“WDR”	account number	withdraw amount				
2: false	“WDR”	account number	withdraw amount	T3	“WDR”	account number	withdraw amount
3: true	“WDR”	account number	withdraw amount	T4	“WDR”	account number	withdraw amount
3: false	“WDR”	account number	withdraw amount				

Notes for the table and code:
-i in master_list element in trans_list reading from *MasterAccountsFile.txt*
-line: a string of combination of a key and its value, e.g. “2233445 100000 Kelly”

Decision	i in trans_list input	line input	Test	i	line
4: true	a list of one line transatcion items separated by space	string of one line transaction	T5	a list of one line transatcion items separated by space	string of one line transaction
4: false	a list of one line transatcion items separated by space	string of one line transaction	T6	a list of one line transatcion items separated by space	string of one line transaction

3) Creation of Test Inputs for Test Cases

		input based on code			input from file		expected output		
Test	Purpose	i[0] input	i[1] input	i[2] input	input from *TransactionSummaryFile.txt*	input from *MasterAccountsFile.txt*	expected *MasterAccountsFile.txt*	expected *ValidAccountsList.txt*	expected terminal output
T1	withdraw 100000 cents from a valid account successfully	“WDR”	“1234567”	“100000”	'WDR 1234567 100000 0000000 Theo\n', 'EOS 0000000 000 0000000 ***'	"1234567 200000 Theo\n"	"1234567 100000 Theo\n"	“1234567\n” “0000000”	“New Master Accounts File created successfully!” “New Valid Accounts File created successfully!”
T2	do transactions	“DEP”	“1234567”	“200000”	'DEP 1234567 100000 0000000	"1234567 200000	"1234567 300000	“1234567\n” “0000000”	“New Master Accounts File

	other than withdraw on a valid account successfully (deposit 100000 cents succesfully)				Theo\n', 'EOS 0000000 000 0000000 ***'	Theo\n"	Theo\n"		created successfully!" "New Valid Accounts File created successfully!"
T3	error when a valid account is not in the *MasterAccountsFile.txt*	"WDR"	"1122334"	"100000"	'WDR 1122334 100000 0000000 Theo\n', 'EOS 0000000 000 0000000 ***'	"1234567 200000 Theo\n"	"1234567 200000 Theo\n"	"1234567\n" "0000000"	'Error! 1122334 not in the master account file!'
T4	error when withdraw over the balance(withdraw 300000 cents from an account has only 100000 cents)	"WDR"	"1234567"	"300000"	'WDR 1234567 300000 0000000 Theo\n', 'EOS 0000000 000 0000000 ***'	"1234567 100000 Theo\n"	"1234567 100000 Theo\n"	"1234567\n" "0000000"	"Error! 1234567 have a negative balance!"

		input based on code			input from file		expected output	
Test	Purpose	i	line	input from *TransactionSummaryFile.txt*	input from *MasterAccountsFile.txt*	expected *MasterAccountsFile.txt*	expected *ValidAccountsList.txt*	expected Terminate output
T5	Error when one line in "TransactionSummaryFile.txt " exceeds 47 characters (the balance has 23 digits, name has 17 characters with code, account number, spaces and "\n"space, then 50 characters in total)	['WDR', '2233445', '100000', '0000000', 'KellyKellyKelly Ke']	"2233445 1234567 8901234 5678901 23 KellyKellyKellyKe"	'WDR 2233445 100000 0000000 KellyKellyKellyKe\n', 'EOS 0000000 000 0000000 ***'	"2233445 12345678901 23456789012 3 KellyKellyKellyKe\n"	"2233445 12345678901 23456789012 3 KellyKellyKellyKe\n"	"2233445\n" "0000000"	"Error! Each line is at most 47 characters!"
T6	withdraw successfully when each line is within 47 characters(e.g. withdraw 100000	['WDR', '1234567', '100000', '0000000', 'KellyKell	"123456 7 100000 theo"	'WDR 2233445 100000 0000000 KellyKellyKellyKe\n', 'EOS	"2233445 12345678901 23456789 KellyKellyKellyKe\n"	"2233445 12345678901 23356789 KellyKellyKellyKe\n"	"2233445\n" "0000000"	"New Master Accounts File created successfully!" "New Valid

	cents)	yKellyKe'] 		0000000 000 0000000 ***'				Accounts File created successfully!”

4.1) Test Report - Resulting of the testing

Decision Coverage Method for testing withdraw

Note: if terminal outputs “New Master Accounts File created successfully!” and “New Valid Accounts File created successfully!”, then it means the files are generated successfully

Test	Purpose	input		output		
		input from “TransactionSummaryFile.txt”	input from “MasterAccountsFile.txt”	updated “MasterAccountsFile.txt”	updated “ValidAccountsList.txt”	Terminate output
T1	withdraw 100000 cents from a valid account successfully	'WDR 1234567 100000 0000000 Theo\n', 'EOS 0000000 000 0000000 ***'	"1234567 200000 Theo\n"	"1234567 100000 Theo\n"	“1234567\n” “0000000”	“New Master Accounts File created successfully!” “New Valid Accounts File created successfully!”
T2	do transactions other than withdraw on a valid account successfully(Deposit 100000 cents succesfully)	'DEP 1234567 100000 0000000 Theo\n', 'EOS 0000000 000 0000000 ***'	"1234567 200000 Theo\n"	"1234567 300000 Theo\n"	“1234567\n” “0000000”	“New Master Accounts File created successfully!” “New Valid Accounts File created successfully!”
T3	error when the account is not in the “MasterAccountsFile.txt”	'WDR 1122334 100000 0000000 Theo\n', 'EOS 0000000 000 0000000 ***'	"1234567 200000 Theo\n"	"1234567 200000 Theo\n"	“1234567\n” “0000000”	'Error! 1122334 not in the master account file!'
T4	error when withdraw over the balance(e.g. withdraw 300000 cents from an account has only 100000cents)	'WDR 1234567 300000 0000000 Theo\n', 'EOS 0000000 000 0000000 ***'	"1234567 100000 Theo\n"	"1234567 100000 Theo\n"	“1234567\n” “0000000”	"Error! 1234567 have a negative balance!"
T5	error when one line in “TransactionSummaryFile.txt ” exceeds 47 characters (e.g. the balance has 23 digits, name has 17 characters with code, account number, spaces and “\n”space, then 50 characters in total)	'WDR 2233445 100000 0000000 KellyKellyKelly Ke\n', 'EOS 0000000 000 0000000 ***'	"2233445 12345678901234567890123 KellyKellyKelly Ke\n"	"2233445 123456789012345 67890123 KellyKellyKellyKe\n"	“2233445\n” “0000000”	"Error! Each line is at most 47 characters!"
T6	withdraw successfully when each line is within 47	'WDR 2233445 100000 0000000 KellyKellyKelly	"2233445 1234567890123 456789	"2233445 123456789012335 6789	“2233445\n” “0000000”	“New Master Accounts File created successfully!”

	characters(e.g. withdraw 100000 cents)	Ke\n', 'EOS 0000000 000 0000000 ***'	KellyKellyKelly Ke\n"	KellyKellyKellyK e\n"		“New Valid Accounts File created successfully!”

4.2) Failure Report

Test name	Purpose (what it was testing)	Nature of Failure (How the output is wrong)	Error in code	How to Fix
T4	error when withdraw over the balance(e.g. withdraw 300000 cents from an account has only 100000cents)	when input “TransactionSummaryFile.txt” and “MasterAccountsFile.txt”, the expected output message is: “'Error! 1122334 not in the master account file!” 'which is different from terminal: failure (“Error!1122334not in the master account file!”)	In *Backend.py*, there are two missing spaces when print error message for withdraw	In *Backend.py*, add space to the printing statement in the get_new_master_accounts( )