

# Unit Testing

<u>Test Type</u>	<u>Test name</u>	<u>Intention</u>
Loop and decision Coverage for read_old_bank_ accounts from read.py	test_valid_account_successfully_read	Applies positive loop and decision coverage and checks if valid account details are read successfully
	test_invalid_length_throws_error	Negative decision coverage and confirms invalid account details lengths are caught (!= 45)
	test_invalid_account_number_throws_error	Negative decision coverage and confirms invalid account number type is caught
	test_invalid_status_throws_error	Negative decision coverage and confirms invalid status type is caught
	test_invalid_negative_balance_throws_error	Negative decision coverage and confirms negative balances are not allowed
	test_invalid_balance_format_throws_error	Negative decision coverage and confirms incorrect balance format is caught
	test_invalid_transactions_throws_error	Negative decision coverage and confirms invalid transaction type is caught
	test_invalid_plan_type_throws_error	Negative decision coverage and confirms invalid plan type is caught, (!= SP or NP)
	test_invalid_negative_transactions_throws_error	Negative decision coverage and confirms that transactions are not negative
	tets_multiple_avalid_and_invalid_lines_reads_successfully_and_throws_error	Negative and Positive decision and loop coverage, ensures that only valid accounts are processed

	test_empty_file	Provides loop coverage for entering the loop 0 times
Statement Coverage for daily_cost_per_plan from transaction_processor.py	test_daily_cost_empty_file	Verify that an empty file returns zero cost (SP: 0, NP: 0) and ensures the loop runs 0 times.
	test_daily_cost_sp_only	Check that lines ending in SP increment the SP cost branch correctly (and do not affect NP cost).
	test_daily_cost_np_only	Check that lines ending in NP increment the NP cost branch correctly (and do not affect SP cost).
	test_daily_cost_mixed_plans	Ensure that lines ending in SP or NP increment both costs appropriately, and that unrelated endings do not increment cost.
	test_daily_cost_file_not_found	Confirm that if a nonexistent file is passed, the method logs the error and quits the program.
	test_daily_cost_general_exception	Verify handling of unexpected errors (e.g., passing a directory instead of a file) by logging them and returning the default dict.

# Failure Test Log

Test name	Intention	Reason For failure from output	Error In Code	Steps taken to fix error
Run # 1 Main errors were syntactic and easily fixed				
test_valid_account_successfully_read	checks if valid account details are read successfully	Invalid Account details length	Update account length to be equal to 45	Fixed formatting issues
test_invalid_status_throws_error	Negative decision coverage and confirms invalid status type is caught	Assertion mismatch	Missing a ' in the assert statement which was there in the original one	Added the '
test_daily_cost_sp_only	Check that lines ending in SP increment the SP cost branch correctly (and do not affect NP cost).	Assertion mismatch.	Actual output has a different format and has not been rounded to 2 decimal places.	Made sure it was rounded to 2 decimal places by calling round(current+0.05, 2). Also changed NP.
test_daily_cost_file_not_found	Verify that an empty file returns zero cost (SP: 0, NP: 0) and ensures the loop runs 0 times.	Attribute Error	Didn't use self.logger, it was TransactionProcessor.logger	Changed to self.logger
test_daily_cost_general_exception	Verify handling of unexpected errors (e.g., passing a directory instead of a file) by logging them	Attribute Error	Didn't use self.logger, it was TransactionProcessor.logger	Changed to self.logger

	and returning the default dict.			
Run #2				
All daily_cost_per_plan tests	General	TypeError	I need to create an instance of transaction processor first.	Created one for each test case, called that instance instead of generic transaction processor
Run #3				
test_daily_cost_file_not_found	(OLD DEF) Verify that an empty file returns zero cost (SP: 0, NP: 0) and ensures the loop runs 0 times.	Attribute Error	File.close will never work if the path doesn't exist	Changed the test case so it suits what's actually happening with a new definition. Changes to logger were also required, so it stops the program.
test_daily_cost_general_exception	(OLD DEF) Verify handling of unexpected errors (e.g., passing a directory instead of a file) by logging them and returning the default dict.	Attribute Error	File.close will never work if the path doesn't exist	Changed the test case so it suits what's actually happening with a new definition. Changes to logger were also required, so it stops the program.