Code Explanation Findings — explain-code-V2 v1.0.0

File explained: calculator.py
Test file: tests/test_calculator.py
Date: 2026-02-20

Analogy

The calculator works like a **diner counter with a short-order cook**: the menu board (show_menu) tells you what you can order; the cashier (get_number) collects your numbers and rejects invalid input; the cook (add / subtract) does the math and calls out the result; the counter stays open until you say 'q' to quit.

Architecture Summary

Function	Role	Key Behavior
`add(a, b)`	`Pure math`	`Returns a + b`
`subtract(a, b)`	`Pure math`	`Returns a - b`
`get_number(prompt)`	`Input guard`	`Loops until valid float entered`
`show_menu()`	`UI display`	`Prints available options`
`main()`	`Control loop`	`Orchestrates the full interaction`

Step-by-Step Walkthrough

add(a, b) — line 1

Pure function. Returns a + b. No side effects.

subtract(a, b) — line 5

Pure function. Returns a - b. No side effects.

get_number(prompt) — line 9

Runs an infinite while True loop. Attempts float(user_input) inside a try/except. If conversion succeeds, the value is returned. If a ValueError is raised (non-numeric input or empty string), an error message is printed and the loop retries.

show_menu() — line 18

Prints the menu to stdout. No inputs, no return value. Called at the top of every iteration of the main loop.

main() — line 26

Outer while True loop. Each iteration: calls show_menu(), reads the user's choice, then routes to the correct branch: 'q' → prints Goodbye! and breaks; '+' or '-' → gets two numbers, computes, prints result; anything else → prints Invalid option.

if __name__ == '__main__' — line 50

Guards the main() call so it only runs when the file is executed directly, not when imported by tests.

Gotcha

get_number() always returns a float. Even if the user types '3', it comes back as 3.0. Output always looks like `3.0 + 4.0 = 7.0`. This is intentional — both integers and decimals work through one code path — but beginners often expect whole numbers to stay whole. The test at test_calculator.py:111 explicitly asserts the float format.

Test Coverage Summary

Test	What it verifies
`test_add_* (4 tests)`	Integers, floats, negatives, large numbers
`test_subtract_* (4 tests)`	Integers, negative results, negatives, large numbers
`test_get_number_valid`	Happy path: valid integer input
`test_get_number_float`	Happy path: decimal input
`test_get_number_invalid_then_valid`	Error recovery loop
`test_get_number_empty_string_then_valid`	Empty input recovery
`test_show_menu_output`	Menu text present in stdout
`test_main_quit`	'q' exits cleanly
`test_main_addition`	Full addition flow
`test_main_subtraction`	Full subtraction flow
`test_main_invalid_option`	Invalid choice shows error message

Total: 15 tests covering all functions and edge cases.