EX.NO.: 06
DATE: 21.07.2025

SELENIUM FOR CALCULATOR

AIM

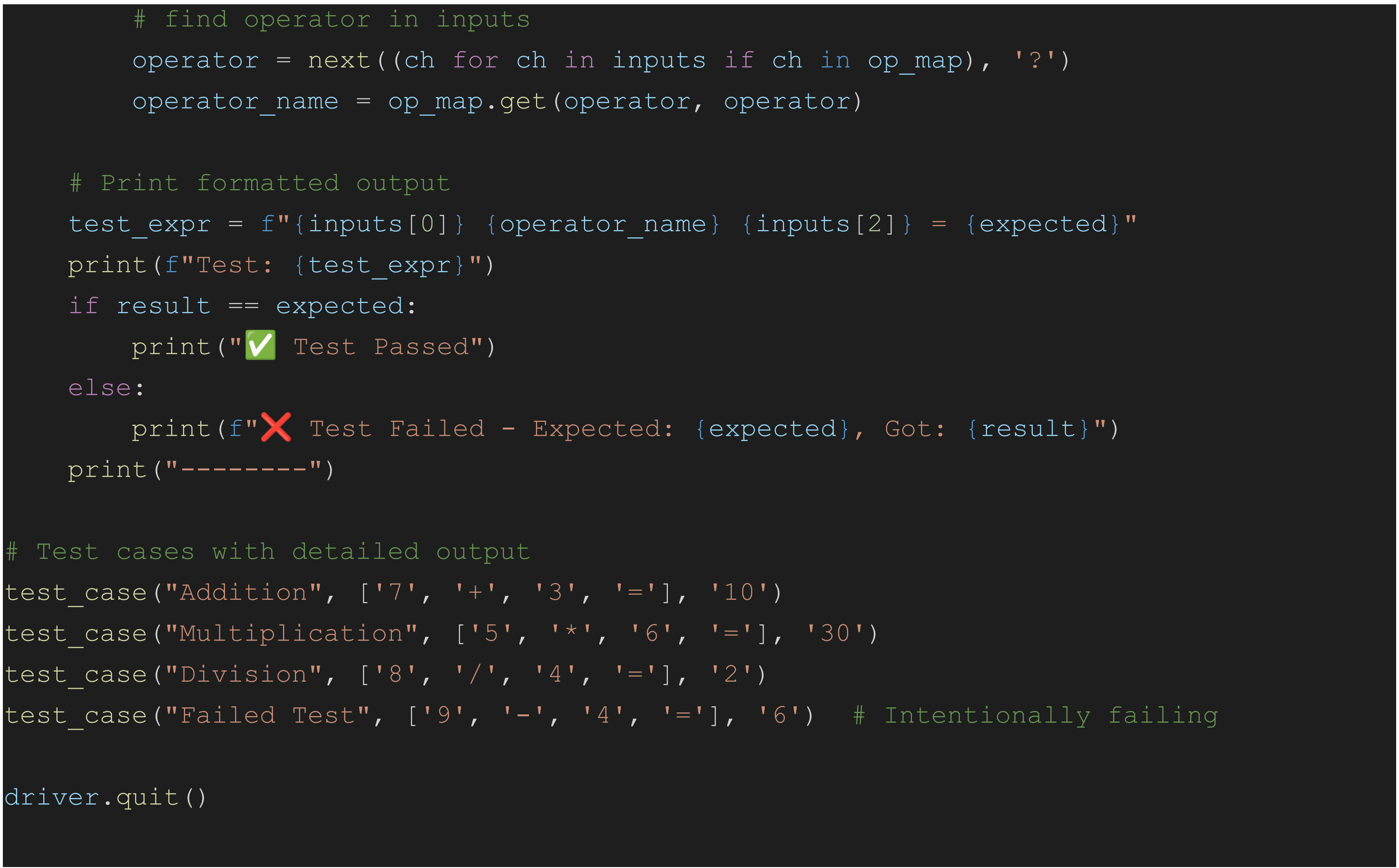
To automate the testing of a basic calculator web interface using Selenium WebDriver in Python, validating the correctness of arithmetic operations (addition, multiplication, division, subtraction) by simulating user input and verifying displayed results.

ALGORITHM

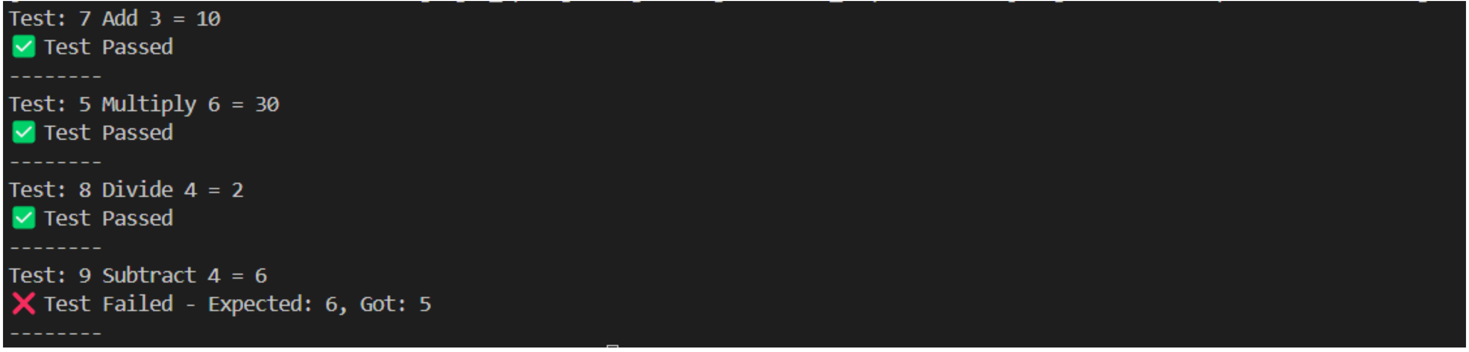
1. **Setup**: Initialize the Selenium WebDriver and open the local calculator HTML file.
2. **Test Execution**: For each arithmetic test case:
3. Clear any previous input or result by clicking the "C" (clear) button.
4. Click calculator buttons corresponding to the operands, operator, and equals sign (=).
5. Retrieve the displayed result from the calculator's output field.
6. **Verification**:
    a. Compare the retrieved result against the expected output.
    b. Print detailed, human-readable test outcome with pass/fail status.
7. **Cleanup**: Close the browser window after all tests complete.

CODE AND OUTPUT

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

# Set up the browser
driver = webdriver.Chrome()

# IMPORTANT: Use file:/// prefix for local files and fix backslashes or spaces if any
in your path
file_path = "D:/TARU/V th year/Software Testing lab/Ex 6/calculator.html"
driver.get("file:///" + file_path.replace("\\", "/"))

def click_buttons(buttons):
    for btn in buttons:
        driver.find_element(By.XPATH, f"//button[text()='{btn}']").click()
    time.sleep(1)

def get_result():
    return driver.find_element(By.ID, "display").get_attribute("value")

def test_case(description, inputs, expected, operator_name=None):
    click_buttons(['C'])  # Clear before test
    click_buttons(inputs)
    result = get_result()

    # Build descriptive operator name if not passed
    if not operator_name:
        # map operator symbol to words
        op_map = {'+': 'Add', '-': 'Subtract', '*': 'Multiply', '/': 'Divide'}
```



```python
        # find operator in inputs
        operator = next((ch for ch in inputs if ch in op_map), '?')
        operator_name = op_map.get(operator, operator)

    # Print formatted output
    test_expr = f"{inputs[0]} {operator_name} {inputs[2]} = {expected}"
    print(f"Test: {test_expr}")
    if result == expected:
        print("✅ Test Passed")
    else:
        print(f"❌ Test Failed - Expected: {expected}, Got: {result}")
    print("--------")

# Test cases with detailed output
test_case("Addition", ['7', '+', '3', '='], '10')
test_case("Multiplication", ['5', '*', '6', '='], '30')
test_case("Division", ['8', '/', '4', '='], '2')
test_case("Failed Test", ['9', '-', '4', '='], '6')  # Intentionally failing

driver.quit()
```



```
Test: 7 Add 3 = 10
✅ Test Passed
--------
Test: 5 Multiply 6 = 30
✅ Test Passed
--------
Test: 8 Divide 4 = 2
✅ Test Passed
--------
Test: 9 Subtract 4 = 6
❌ Test Failed - Expected: 6, Got: 5
--------
```

INFERENCE

The Selenium script effectively automates testing of basic calculator operations by simulating user inputs and verifying results. It accurately detects both correct and incorrect outcomes, ensuring reliable validation of the calculator's functionality.