Patrick McCabe
David Thompson

Sugar Labs: Calculate Activity Automated Testing Framework

CSCI 362-01
Fall 2020
Dr. Bowring

Introduction

The objective of this assignment was to design and implement an automated testing framework for an HFOSS project of our choice. We selected Sugar Labs, a Python-based project that is aimed at helping children learn through a simplified operating system and a variety of activities. Unfortunately, due to dependency issues, we could not work with the entire project, and instead focused on the project's Calculate activity.

The Framework

Test Cases: Each test to be run is recorded in a formatted text file that contains critical information about the test: a test ID, what requirement is being tested, which component and function are being tested, what driver must be used for the test, what inputs are used for the test, and the oracle, or expected output, for the test.

```
Test Suite: TS0001
Test ID: TC0001
Requirement: Addition of two whole numbers
Component: functions.py
Driver: testAdd.py
Method: add(x, y)
Inputs: 2, 1
Oracle: 3
```

Drivers: Each method is testing using a unique driver. Drivers call accept inputs from a test case and return an output by running the inputs through the desired method in the source code.

```python
#!/bin/env python3

# import necessary libraries
import math
import sys

# Script Name: testAdd.py
# Method: testAdd(operands)
# Purpose: test the addition of two numbers for
# the calculate-activity
# Inputs: list of operands
# Outputs: the result of the add(x, y) method

def testAdd(operands):

    sys.path.append('project/src/calculate-activity')

    import functions as fun


    # invoke the add method for the functions.py file
    test = fun.add(operands[0], operands[1])

    # return the result
    return test
```

Scripts: The framework is run by a series of scripts:

runAllTests: Called from the command line to begin the testing process. Utilizes the helper scripts to perform the desired tasks.

parser: Reads through test case files and converts information into a format useable by the other scripts. Called by runAllScripts.

tester: Selects driver based on current test case, feeds in appropriate inputs, runs tested method, compares output to test case oracle, and determines if the test passed or failed. Called by runAllScripts.

report: Uses information from test cases and results from tester to create a report of the test results in an html document. Called by runAllScripts.

Fault Injection

After completing our framework, we injected a series of faults into our source code to see how the testing results would be altered. Ideally, this would cause most, but not all, of our tests to fail. One fault was inserted into each method tested:

add: replaced a + with a -

sub: replaced a = with a +

mul: added a *

div: added a /

pow: removed a *

Outcomes

Throughout this project, we had the opportunity to learn about and improve our skills with Git, Bash, Python, Linux, and the command line terminal. We got to practice working on a development project in a team environment. We learned about the practice and importance of software testing. In the end, we were happy with the results of our work, but can also identify areas where out framework could be improved; these potential improvements gave us a lesson in the difference between successful implementation of a design and the quality of the design itself.

Calculate Activity Test Report 2020-11-12 09:06:48

Test ID	Requirement	Component	Driver	Method	Inputs	Oracle	Output	Result
TC0001	Addition of two numbers	functions.py	testAdd.py	add(x, y)	[1, 2]	3	3	Test Passed
TC0002	Addition of two numbers	functions.py	testAdd.py	add(x, y)	[-1, 1]	0	0	Test Passed
TC0003	Addition of two numbers	functions.py	testAdd.py	add(x, y)	[1.1, 2.1]	3.2	3.2	Test Passed
TC0004	Addition of two numbers	functions.py	testAdd.py	add(x, y)	[3.141592653589793, 1.4142135623730951]	4.555806215962888	4.555806215962888	Test Passed
TC0005	Addition of two numbers	functions.py	testAdd.py	add(x, y)	[9223372036854775807, 2]	9223372036854775809	9223372036854775809	Test Passed
TC0006	Subtraction of two numbers	functions.py	testSub.py	sub(x, y)	[1, 2]	-1	-1	Test Passed
TC0007	Subtraction of two numbers	functions.py	testSub.py	sub(x, y)	[-1, 1]	-2	-2	Test Passed
TC0008	Subtraction of two numbers	functions.py	testSub.py	sub(x, y)	[1.1, 2.1]	-1	-1.0	Test Passed
TC0009	Subtraction of two numbers	functions.py	testSub.py	sub(x, y)	[3.141592653589793, 1.4142135623730951]	1.727379091216698	1.727379091216698	Test Passed
TC0010	Subtraction of two numbers	functions.py	testSub.py	sub(x, y)	[9223372036854775807, 18446744073709551616]	-9223372036854775809	-9223372036854775809	Test Passed
TC0011	Multiplication of two numbers	functions.py	testMul.py	mul(x, y)	[1, 2]	2	2	Test Passed
TC0012	Multiplication of two numbers	functions.py	testMul.py	mul(x, y)	[-1, 1]	-1	-1	Test Passed
TC0013	Multiplication of two numbers	functions.py	testMul.py	mul(x, y)	[1.1, 2.1]	2.3100000000000005	2.3100000000000005	Test Passed
TC0014	Multiplication of two numbers	functions.py	testMul.py	mul(x, y)	[3.141592653589793, 1.4142135623730951]	4.442882938158366	4.442882938158366	Test Passed
TC0015	Multiplication of two numbers	functions.py	testMul.py	mul(x, y)	[9223372036854775807, 2]	18446744073709551614	18446744073709551614	Test Passed
TC0016	Division of two numbers	functions.py	testDiv.py	div(x, y)	[1, 2]	0.5	1/2	Test Failed
TC0017	Division of two numbers	functions.py	testDiv.py	div(x, y)	[-1, 1]	-1.0	-1	Test Failed
TC0018	Division of two numbers	functions.py	testDiv.py	div(x, y)	[1.1, 2.1]	0.5238095238095238	0.5238095238095238	Test Passed
TC0019	Division of two numbers	functions.py	testDiv.py	div(x, y)	[3.141592653589793, 1.4142135623730951]	2.221441469079183	2.221441469079183	Test Passed
TC0020	Division of two numbers	functions.py	testDiv.py	div(x, y)	[9223372036854775807, 2]	4.611686018427388e+18	4.611686018427388e+18	Test Passed
TC0021	Exponentiation of two numbers	functions.py	testPow.py	pow(x, y)	[1, 2]	1	1	Test Passed
TC0022	Exponentiation of two numbers	functions.py	testPow.py	pow(x, y)	[-1, 1]	-1	-1	Test Passed
TC0023	Exponentiation of two numbers	functions.py	testPow.py	pow(x, y)	[1.1, 2.1]	1.2215876651600335	1.2215876651600334	Test Failed
TC0024	Exponentiation of two numbers	functions.py	testPow.py	pow(x, y)	[3.141592653589793, 1.4142135623730951]	5.047497267370911	5.047497267370911089	Test Failed
TC0025	Exponentiation of two numbers	functions.py	testPow.py	pow(x, y)	[9223372036854775807, 2]	85070591730234615847396907784232501249	85070591730234615847396907784232501249	Test Passed

Calculate Activity Test Report 2020-11-21 20:39:12

Test ID	Requirement	Component	Driver	Method	Inputs	Oracle	Output	Result
TC0001	Addition of two whole numbers	functions.py	testAdd.py	add(x, y)	[2, 1]	3	1	Test Failed
TC0002	Addition of negative and positive integer	functions.py	testAdd.py	add(x, y)	[-1, 1]	0	-2	Test Failed
TC0003	Addition of two floats	functions.py	testAdd.py	add(x, y)	[1.1, 2.1]	3.2	-1.0	Test Failed
TC0004	Addition of two real numbers	functions.py	testAdd.py	add(x, y)	[3.141592653589793, 1.4142135623730951]	4.555806215962888	1.727379091216698	Test Failed
TC0005	Addition involving max sized number	functions.py	testAdd.py	add(x, y)	[9223372036854775807, 2]	9223372036854775809	9223372036854775805	Test Failed
TC0006	Subtracion of two whole numbers	functions.py	testSub.py	sub(x, y)	[1, 2]	-1	3	Test Failed
TC0007	Subtraction of negative and positive integer	functions.py	testSub.py	sub(x, y)	[-1, 1]	-2	0	Test Failed
TC0008	Subtraction of two floats	functions.py	testSub.py	sub(x, y)	[1.1, 2.1]	-1	3.2	Test Failed
TC0009	Subtraction of two real numbers	functions.py	testSub.py	sub(x, y)	[3.141592653589793, 1.4142135623730951]	1.727379091216698	4.555806215962888	Test Failed
TC0010	Subtraction involving max sized number	functions.py	testSub.py	sub(x, y)	[9223372036854775807, 18446744073709551616]	-9223372036854775809	27670116110564327423	Test Failed
TC0011	Multiplication of two whole numbers	functions.py	testMul.py	mul(x, y)	[2, 1]	2	2	Test Passed
TC0012	Multiplication of negative and positive integer	functions.py	testMul.py	mul(x, y)	[-1, 1]	-1	-1	Test Passed
TC0013	Multiplication of two floats	functions.py	testMul.py	mul(x, y)	[1.1, 2.1]	2.3100000000000005	1.2215876651600335	Test Failed
TC0014	Multiplication of two real numbers	functions.py	testMul.py	mul(x, y)	[3.141592653589793, 1.4142135623730951]	4.442882938158366	5.047497267370911	Test Failed
TC0015	Multiplication involving max sized number	functions.py	testMul.py	mul(x, y)	[9223372036854775807, 2]	18446744073709551614	85070591730234615847396907784232501249	Test Failed
TC0016	Division of two whole numbers	functions.py	testDiv.py	div(x, y)	[2, 1]	2	2	Test Failed
TC0017	Division of positive and negative integer	functions.py	testDiv.py	div(x, y)	[-1, 1]	-1.0	-1	Test Failed
TC0018	Division of two floats	functions.py	testDiv.py	div(x, y)	[1.1, 2.1]	0.5238095238095238	0.0	Test Failed
TC0019	Division of two real numbers	functions.py	testDiv.py	div(x, y)	[3.141592653589793, 1.4142135623730951]	2.221441469079183	2.0	Test Failed
TC0020	Division involving max sized number	functions.py	testDiv.py	div(x, y)	[9223372036854775807, 2]	4.611686018427388e+18	4611686018427387903	Test Failed
TC0021	Exponentiation of two whole numbers	functions.py	testPow.py	pow(x, y)	[2, 1]	2	2	Test Passed
TC0022	Exponentiation of negative and positive integer	functions.py	testPow.py	pow(x, y)	[-1, 1]	-1	-1	Test Passed
TC0023	Exponentiation of two floats	functions.py	testPow.py	pow(x, y)	[1.1, 2.1]	1.2215876651600335	1.2215876651600334	Test Failed
TC0024	Exponentiation of two real numbers	functions.py	testPow.py	pow(x, y)	[3.141592653589793, 1.4142135623730951]	5.047497267370911	5.047497267370911089	Test Failed
TC0025	Exponentiation involving max sized number	functions.py	testPow.py	pow(x, y)	[9223372036854775807, 2]	85070591730234615847396907784232501249	18446744073709551614	Test Failed