

Name: Dominic Gopalakrishnan

ID: 101289239

CRN: 49746

## Full Stack Development

### Lab Exercise 2

Screenshots:

Output:

```
> lab2_mocha_test@1.0.0 test C:\Users\domin\Desktop\lab2_mocha_test
> mocha

Add()
  ✓ 5 + 2 should return 7 and PASS
  1) 5 + 2 should not return 8 and FAIL

Sub()
  ✓ 5 - 2 should return 3 and PASS
  2) 5 - 2 should not return 5 and FAIL

Mul()
  ✓ 5 * 2 should return 10 and PASS
  3) 5 * 2 should not return 12 and FAIL

Div()
  ✓ 10 / 2 should return 5 and PASS
  4) 10 / 2 should not return 2 and FAIL

  4 passing (10ms)
  4 failing
```

Expected and Actual Values:

```
1) Add()
  5 + 2 should not return 8 and FAIL:

AssertionError: expected 7 to equal 8
+ expected - actual

-7
+8

at Context.<anonymous> (test\calculator.js:10:35)
at processImmediate (internal/timers.js:464:21)

2) Sub()
  5 - 2 should not return 5 and FAIL:

AssertionError: expected 3 to equal 5
+ expected - actual

-3
+5

at Context.<anonymous> (test\calculator.js:19:35)
at processImmediate (internal/timers.js:464:21)

3) Mul()
  5 * 2 should not return 12 and FAIL:

AssertionError: expected 10 to equal 12
+ expected - actual

-10
+12

at Context.<anonymous> (test\calculator.js:28:35)
at processImmediate (internal/timers.js:464:21)

4) Div()
  10 / 2 should not return 2 and FAIL:

AssertionError: expected 5 to equal 2
+ expected - actual

-5
+2
```

Source code:

Calculator APP:

```
app > js calculator.js > ...
1  exports.add = function(i, j) {
2    |  ...
3    |  return i + j;
4    |;
5  exports.mul = function(i, j) {
6    |  ...
7    |  return i * j;
8  };
9  exports.div = function(i, j) {
10 |  ...
11 |  return i / j;
12 };
13 exports.sub = function(i, j) {
14 |  ...
15 |  return i - j;
```

test/calculator.js:

```
test > [calculator.js] ...  
1 const expect = require('chai').expect;  
2 const calculator = require('../app/calculator');  
3  
4 describe('Add()', function() {  
5     it('5 + 2 should return 7 and PASS', function() {  
6         expect(calculator.add(5, 2)).to.equal(7);  
7     });  
8     it('5 + 2 should not return 8 and FAIL', function() {  
9         expect(calculator.add(5, 2)).to.equal(8);  
10    });  
11});  
12  
13 describe('Sub()', function() {  
14     it('5 - 2 should return 3 and PASS', function() {  
15         expect(calculator.sub(5, 2)).to.equal(3);  
16     });  
17     it('5 - 2 should not return 5 and FAIL', function() {  
18         expect(calculator.sub(5, 2)).to.equal(5);  
19     });  
20});  
21  
22 describe('Mul()', function() {  
23     it('5 * 2 should return 10 and PASS', function() {  
24         expect(calculator.mul(5, 2)).to.equal(10);  
25     });  
26     it('5 * 2 should not return 12 and FAIL', function() {  
27         expect(calculator.mul(5, 2)).to.equal(12);  
28     });  
29});|  
30  
31 describe('Div()', function() {  
32     it('10 / 2 should return 5 and PASS', function() {  
33         expect(calculator.div(10, 2)).to.equal(5);  
34     });  
35     it('10 / 2 should not return 2 and FAIL', function() {  
36         expect(calculator.div(10, 2)).to.equal(2);  
37     });  
38});  
39
```