Lab 2 Comp 3133

Test 1:

Test 2:

```
2) Calculator Tests
    sub()
    5 - 2 should NOT return 5 (FAIL):

AssertionError: expected 3 to equal 5
+ expected - actual

-3
+5

at Context.<anonymous> (test/calculator.js:23:34)
at processImmediate (node:internal/timers:466:21)
```

Test 3:

```
3) Calculator Tests
mul()
5 * 2 should NOT return 12 (FAIL):

AssertionError: expected 10 to equal 12
+ expected - actual

-10
+12
at Context.<anonymous> (test/calculator.js:34:34)
at processImmediate (node:internal/timers:466:21)
```

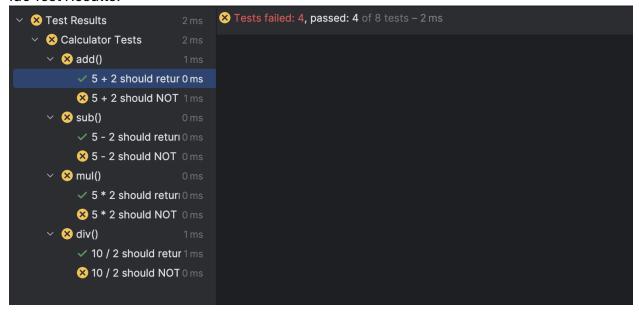
Test 4:

```
4) Calculator Tests
    div()
      10 / 2 should NOT return 2 (FAIL):

AssertionError: expected 5 to equal 2
+ expected - actual

-5
+2
    at Context.<anonymous> (test/calculator.js:45:35)
    at processImmediate (node:internal/timers:466:21)
```

Ide Test Results:



Source Code:

```
const { expect : ExpectStatic } = require('chai');
                                                                                                     4 4 4 2 ^
      const { add, sub, mul, div } = require('../calculator');
      describe( title: 'Calculator Tests', fn: function() : void {
           describe( title: 'add()', fn: function() : void {
6 🚯
7 🕏
               it( title: '5 + 2 should return 7 (PASS)', fn: function() : void {
                   expect(add( i: 5, j: 2)).to.equal( value: 7);
               it( title: '5 + 2 should NOT return 8 (FAIL)', fn: function() : void {
11 👫
                   expect(add( i: 5, j: 2)).to.equal( value: 8); // This test will intentionally fail
           describe( title: 'sub()', fn: function() : void {
17 🕼
18 😘
               it( title: '5 - 2 should return 3 (PASS)', fn: function() : void {
                   expect(sub( i: 5, j: 2)).to.equal( value: 3);
               it( title: '5 - 2 should NOT return 5 (FAIL)', fn: function() : void {
                   expect(sub( i: 5, j: 2)).to.equal( value: 5); // This test will intentionally fail
           // MULTIPLICATION TESTS
28 🚯
           describe( title: 'mul()', fn: function() : void {
               it( title: '5 * 2 should return 10 (PASS)', fn: function() : void {
                   expect(mul( i: 5, j: 2)).to.equal( value: 10);
```

Source Code:

```
it( title: '5 * 2 should NOT return 12 (FAIL)', fn: function() :void {
    expect(mul( i: 5, j: 2)).to.equal( value: 12); // This test will intentionally fail
    });
}

// DIVISION TESTS

describe( title: 'div()', fn: function() :void {
    it( title: '10 / 2 should return 5 (PASS)', fn: function() :void {
    expect(div( i: 10, j: 2)).to.equal( value: 5);
});

it( title: '10 / 2 should NOT return 2 (FAIL)', fn: function() :void {
    expect(div( i: 10, j: 2)).to.equal( value: 2); // This test will intentionally fail
    });
});

**Provided Head of the standard He
```