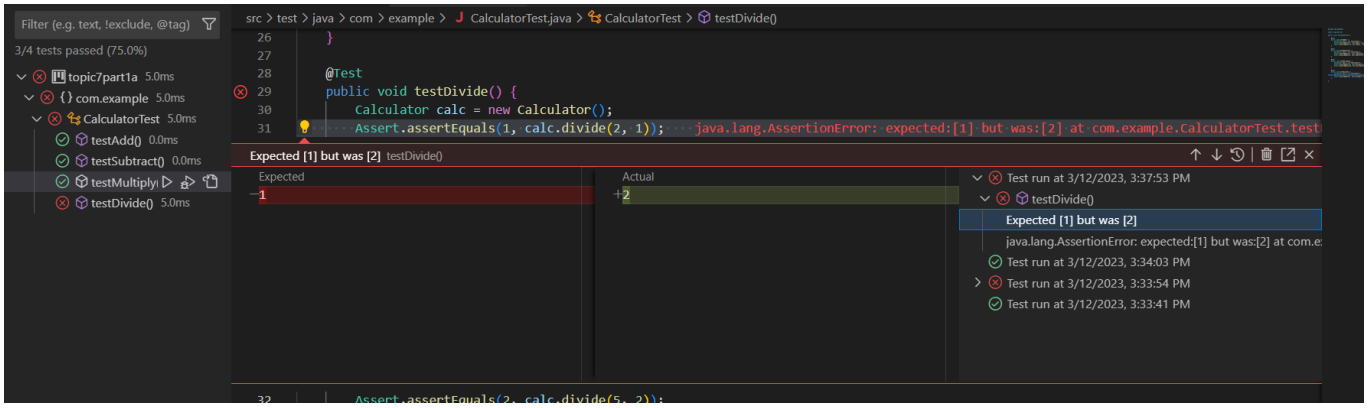
**Activity 7**

Jesse Dalton
March 12, 2023

**PART 1a**

Test Case failure:




**A.**

These are pretty simple functions. The parameters are very generic as they can be any integer and there will almost always be a valid result. I think two tests per function are satisfactory (one for a positive result and one for a negative). The only special case is division by zero. There needs to be a test case that passes 0 as the divisor in the divide function.

**B.**

Black box testing assumes the user or test does not know anything about the internal workings of the program. This type of testing usually checks for functional correctness. White box testing assumes the user or test has access to the internal workings of the program and can be used for testing performance and resource management.

**C.**

You would use a test suite when there are multiple classes to test and therefore multiple test classes. The test suite is a driver for all of these test classes.

**PART 3**

Test Runner output:

```
c:\Users\jmdal\Projects\CST239\topics\topic7\topic7part3> c:
&& cd c:\Users\jmdal\Projects\CST239\topics\topic7\topic7part
3 && cmd /C ""C:\Program Files\Java\jdk-17.0.5\bin\java.exe"
@C:\Users\jmdal\AppData\Local\Temp\cp_cu3p8ihma2iurw4je6adqay
xr.argfile com.example.TestRunner "
Tests Passed
```

**A.**

There is an assertThrows() method available in the JUnit framework that will pass if the argument throws an exception when run with the specified arguments.

**B.**

I think the biggest problem with testing ALL possible errors is that is difficult to anticipate every possible error. Otherwise, we probably wouldn't need tests at all. I think testing is a way to think through the possible errors and easily maintain the code as changes are made to fix the errors that arise. Another difficulty is the fact that not all errors are easily tested. A good example of this is race conditions when multi threading.