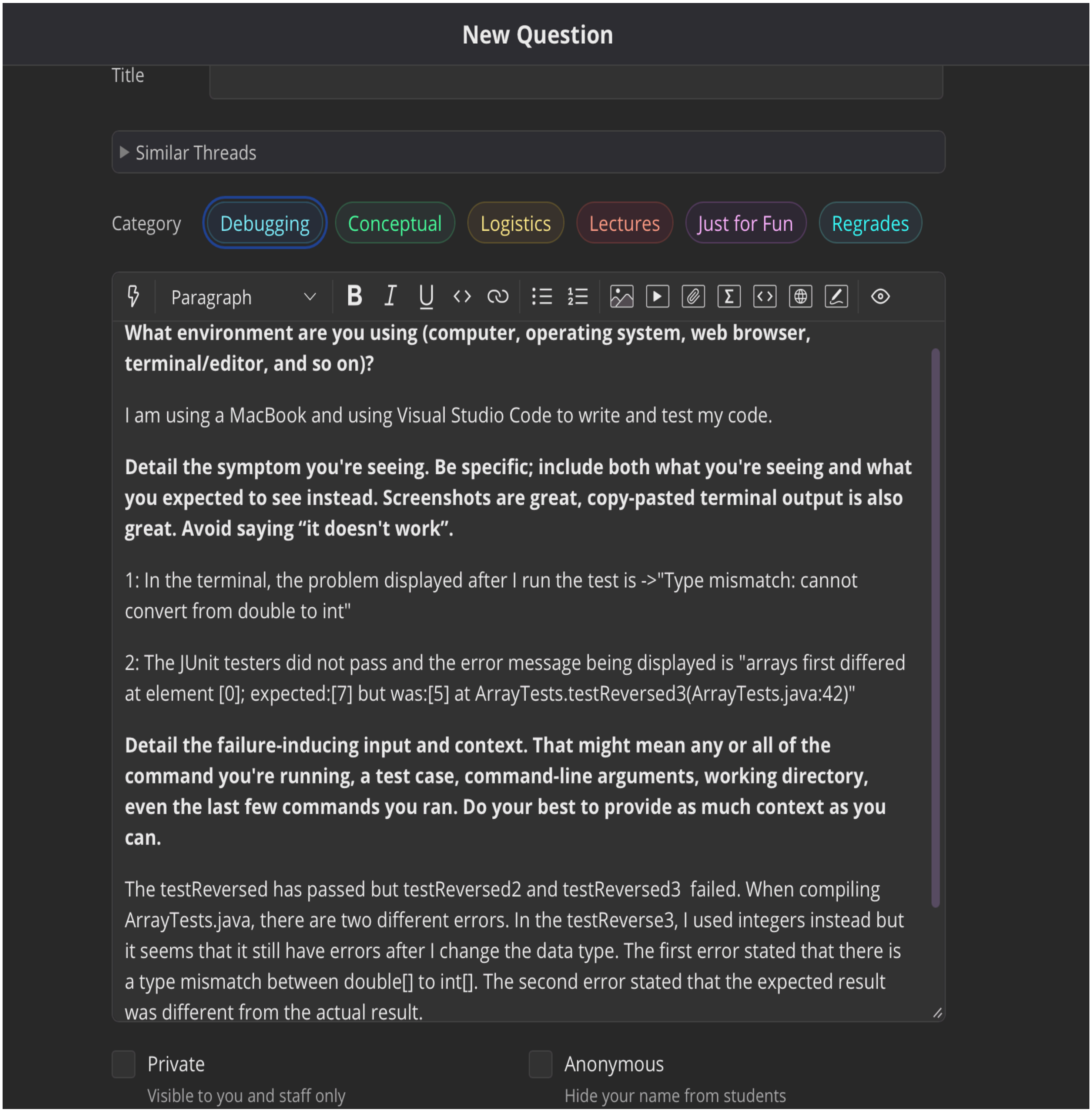
CSE15L-Lab5-Report

Lab Report 5: Debugging and Reflection

This lab report will go over on how to debug an error in a coding scenario and reflecting on the course's lab experience.

Debugging Scenario

Student Post:



What environment are you using (computer, operating system, web browser, terminal/editor, and so on)?

I am using a MacBook and using Visual Studio Code to write and test my code.

Detail the symptom you're seeing. Be specific; include both what you're seeing and what you expected to see instead. Screenshots are great, copy-pasted terminal output is also great. Avoid saying "it doesn't work".

1: In the terminal, the problem displayed after I run the test is ->"Type mismatch: cannot convert from double to int"

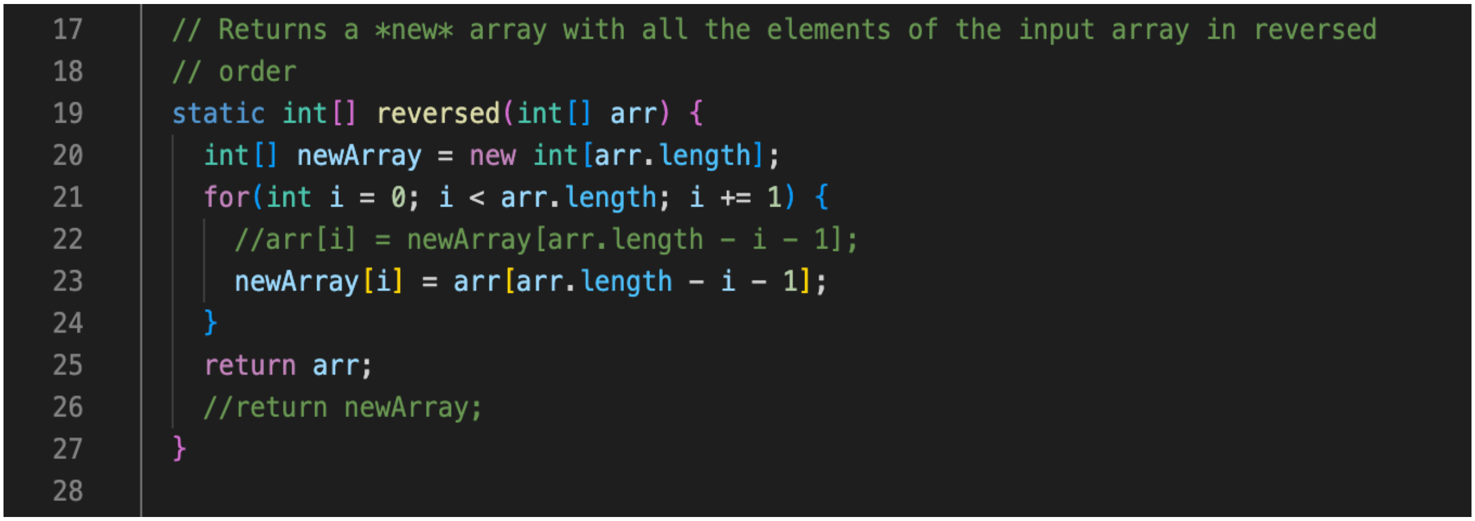
2: The JUnit testers did not pass and the error message being displayed is "arrays first differed at element [0]; expected:[7] but was:[5] at ArrayTests.testReversed3(ArrayTests.java:42)"

Detail the failure-inducing input and context. That might mean any or all of the command you're running, a test case, command-line arguments, working directory, even the last few commands you ran. Do your best to provide as much context as you can.

The testReversed has passed but testReversed2 and testReversed3 failed. When compiling ArrayTests.java, there are two different errors. In the testReverse3, I used integers instead but it seems that it still have errors after I change the data type. The first error stated that there is a type mismatch between double[] to int[]. The second error stated that the expected result was different from the actual result.

Errors/Symptoms

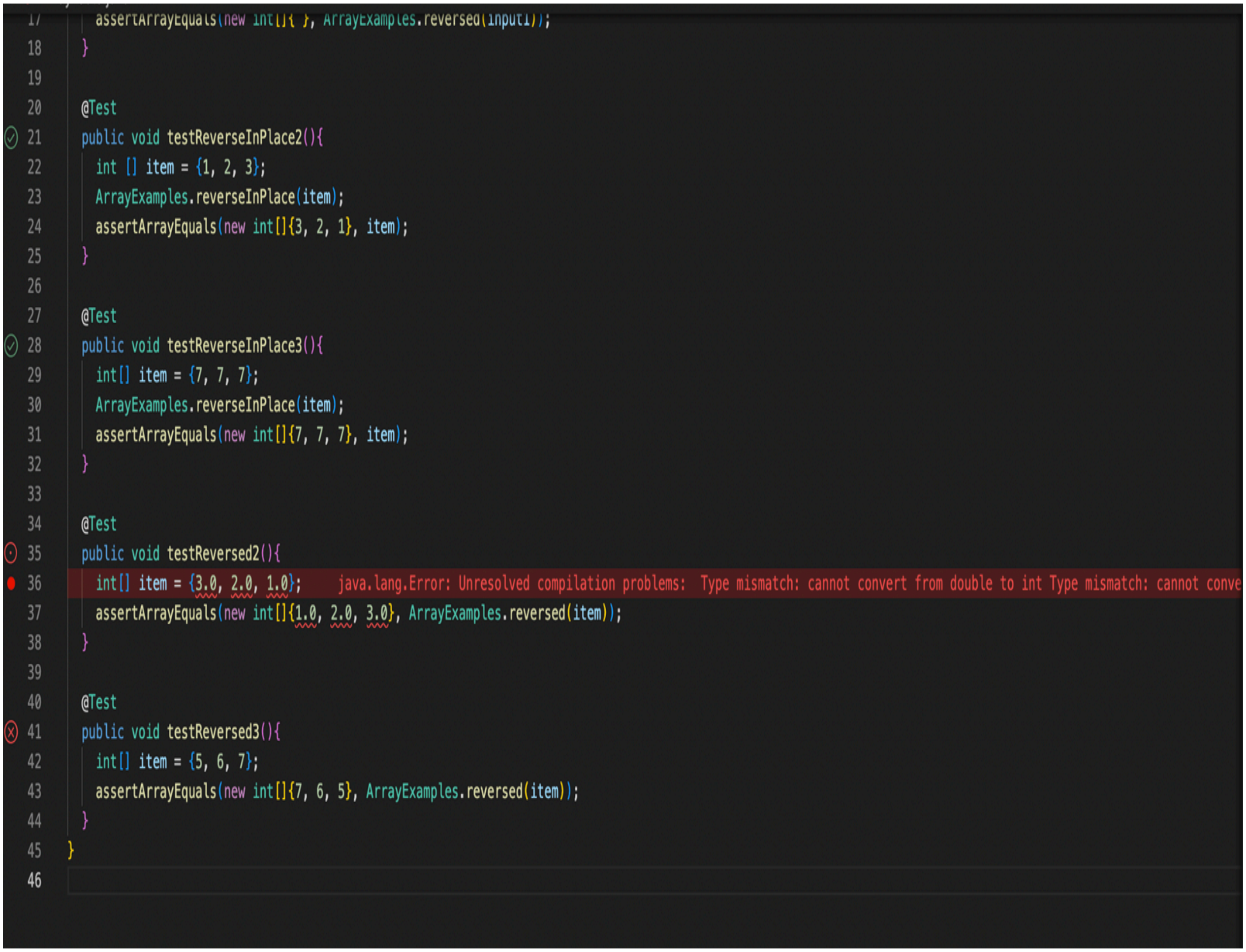
Reversed Method From ArrayExamples.java file



```
  // Returns a *new* array with all the elements of the input array in reversed
  // order
  static int[] reversed(int[] arr) {
    int[] newArray = new int[arr.length];
    for(int i = 0; i < arr.length; i += 1) {
      //arr[i] = newArray[arr.length - i - 1];
      newArray[i] = arr[arr.length - i - 1];
    }
    return arr;
    //return newArray;
  }

```

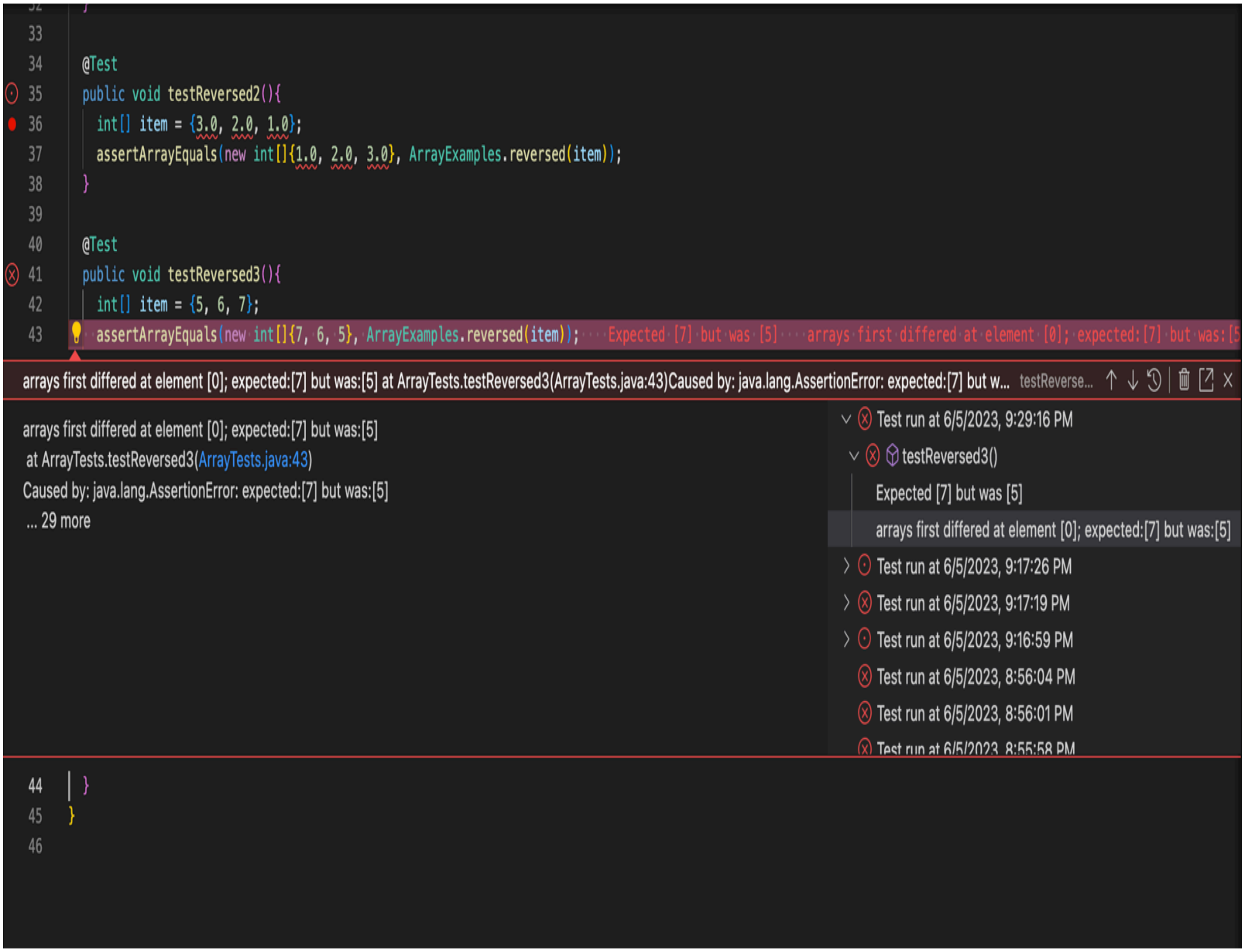
Tests in JUnit

1. Error in testReversed2



```
    assertArrayEquals(new int[]{ }, ArrayExamples.reversed(input1));
  }

  @Test
  public void testReverseInPlace2(){
    int [] item = {1, 2, 3};
    ArrayExamples.reverseInPlace(item);
    assertArrayEquals(new int[]{3, 2, 1}, item);
  }

  @Test
  public void testReverseInPlace3(){
    int[] item = {7, 7, 7};
    ArrayExamples.reverseInPlace(item);
    assertArrayEquals(new int[]{7, 7, 7}, item);
  }

  @Test
  public void testReversed2(){
    int[] item = {3.0, 2.0, 1.0};    java.lang.Error: Unresolved compilation problems:  Type mismatch: cannot convert from double to int Type mismatch: cannot conve
    assertArrayEquals(new int[]{1.0, 2.0, 3.0}, ArrayExamples.reversed(item));
  }

  @Test
  public void testReversed3(){
    int[] item = {5, 6, 7};
    assertArrayEquals(new int[]{7, 6, 5}, ArrayExamples.reversed(item));
  }
}

```

2. Error in testReversed3



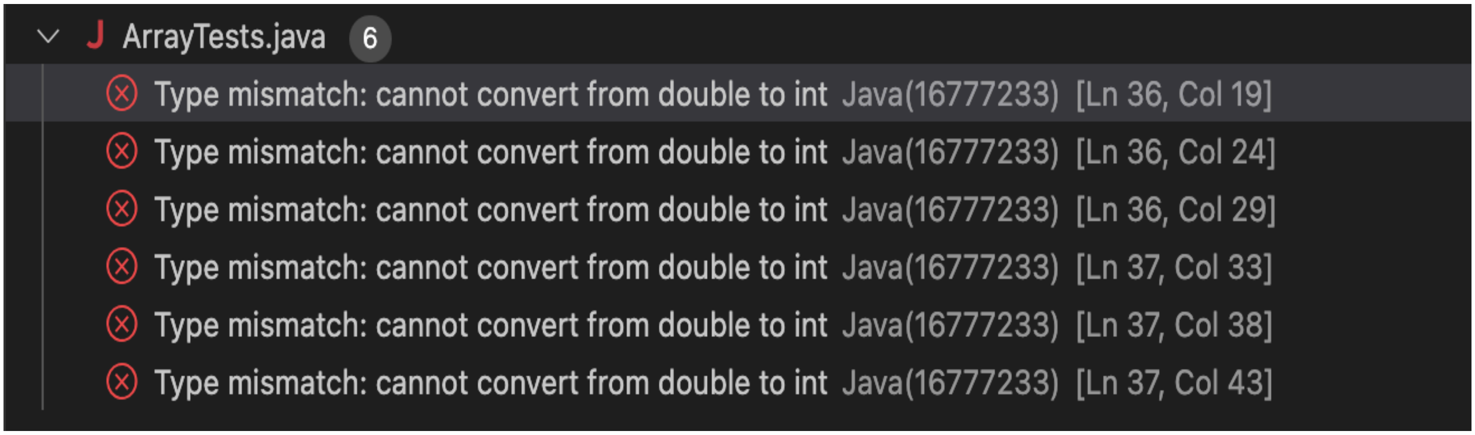
```
33
34    @Test
35    public void testReversed2(){
36      int[] item = {3.0, 2.0, 1.0};
37      assertArrayEquals(new int[]{1.0, 2.0, 3.0}, ArrayExamples.reversed(item));
38    }
39
40    @Test
41    public void testReversed3(){
42      int[] item = {5, 6, 7};
43      assertArrayEquals(new int[]{7, 6, 5}, ArrayExamples.reversed(item));    Expected [7] but was [5]    arrays first differed at element [0]; expected:[7] but was:[5
```

```
arrays first differed at element [0]; expected:[7] but was:[5] at ArrayTests.testReversed3(ArrayTests.java:43)Caused by: java.lang.AssertionError: expected:[7] but w...  testReverse...

arrays first differed at element [0]; expected:[7] but was:[5]
 at ArrayTests.testReversed3(ArrayTests.java:43)
Caused by: java.lang.AssertionError: expected:[7] but was:[5]
 ... 29 more
```

```
Test run at 6/5/2023, 9:29:16 PM
  testReversed3()
    Expected [7] but was [5]
    arrays first differed at element [0]; expected:[7] but was:[5]
Test run at 6/5/2023, 9:17:26 PM
Test run at 6/5/2023, 9:17:19 PM
Test run at 6/5/2023, 9:16:59 PM
Test run at 6/5/2023, 8:56:04 PM
Test run at 6/5/2023, 8:56:01 PM
Test run at 6/5/2023, 8:55:58 PM
```

```
44      }
45    }
46
```

Problem: Error Message in JUnit Testing



```
ArrayTests.java  6
  Type mismatch: cannot convert from double to int  Java(16777233)  [Ln 36, Col 19]
  Type mismatch: cannot convert from double to int  Java(16777233)  [Ln 36, Col 24]
  Type mismatch: cannot convert from double to int  Java(16777233)  [Ln 36, Col 29]
  Type mismatch: cannot convert from double to int  Java(16777233)  [Ln 37, Col 33]
  Type mismatch: cannot convert from double to int  Java(16777233)  [Ln 37, Col 38]
  Type mismatch: cannot convert from double to int  Java(16777233)  [Ln 37, Col 43]
```

Output In the Terminal:

<img src=”junit1.png” width=”700”

Student’s Post Analysis

After observing the code and screenshots of the student’s error, the symptom, bug, and failure-inducing inputs are found.

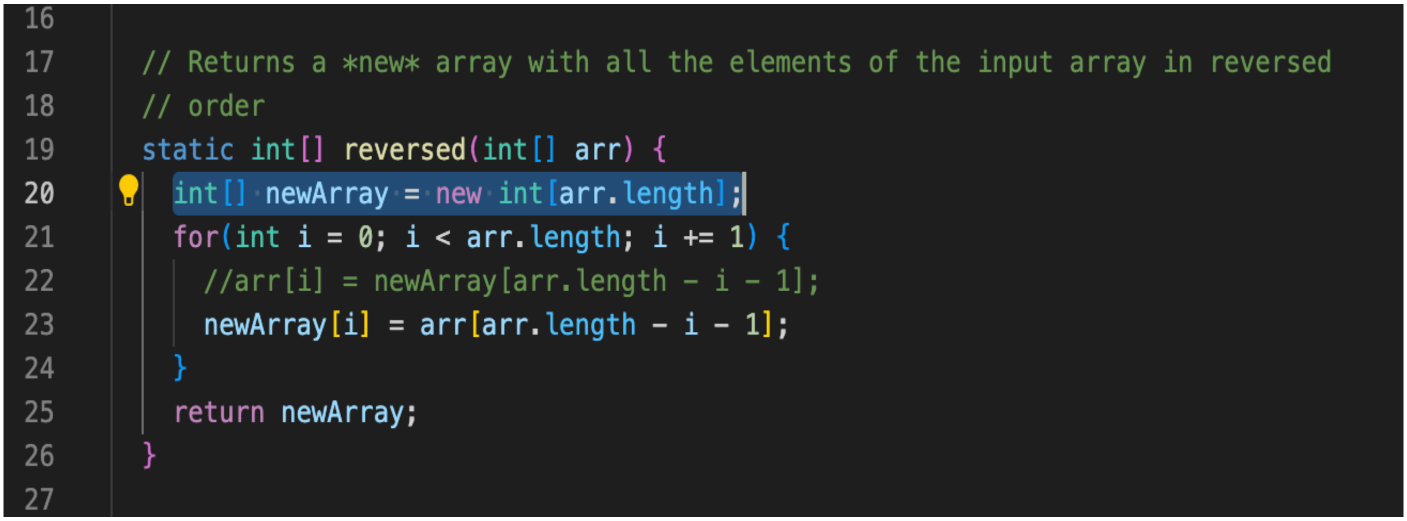
- □**Symptom:** 1: There is a type mismatch as the method reversed cannot convert the data input from a double[] to int[]. 2: The expected output does not match the actual output after the method reversed is being called.
- □**Bug:** The bug stems from line 25.
- □**Failure-Inducing Input:** comes from the tester file of JUnit in the testReversed 2 and testReversed3.

Response from a TA:

The Problem: After checking your reversed method in ArrayExamples.java, it is true that this method takes in inputs in the type of an integer array only as your parameter has stated `int[] arr`. In addition, your return statement is returning the wrong array in line 25 within the reversed method.

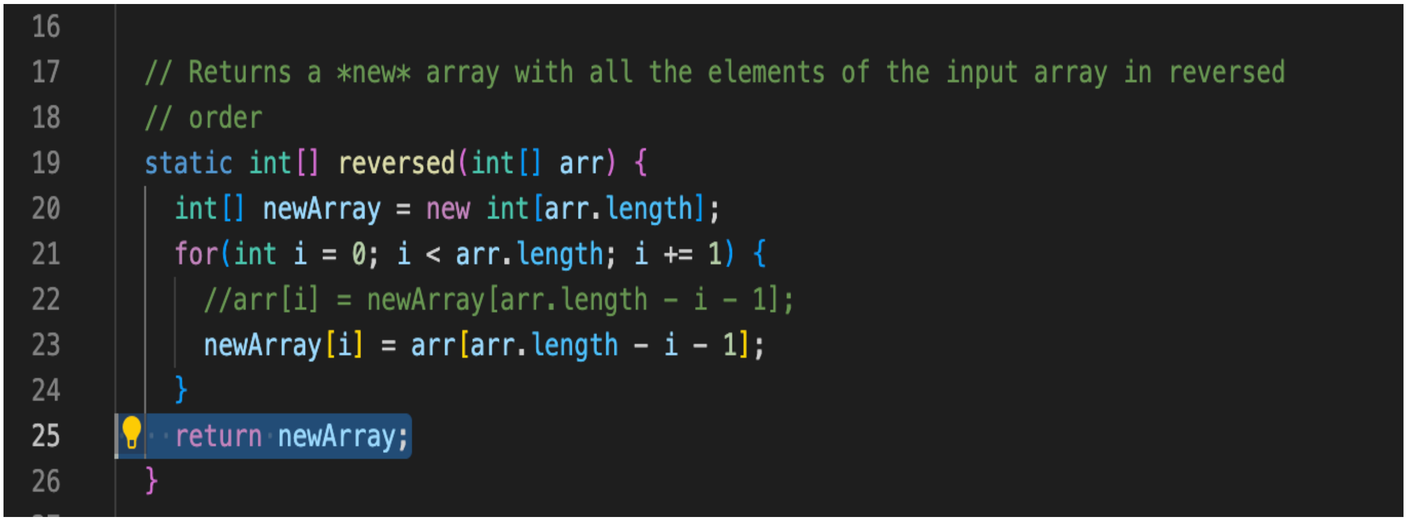
Resolution:

1. **First Error:** you have passed in the wrong argument for reversed method. Because your reversed method in ArrayExamples.java stated that the parameter is `int[]` as its type, the elements of `{3.0, 2.0, 1.0}` are all doubles. Instead, try `{3, 2, 1}` and make sure to also change the element type in your expected value when calling assertArrayEquals.



```
16
17    // Returns a *new* array with all the elements of the input array in reversed
18    // order
19    static int[] reversed(int[] arr) {
20      int[] newArray = new int[arr.length];
21      for(int i = 0; i < arr.length; i += 1) {
22        //arr[i] = newArray[arr.length - i - 1];
23        newArray[i] = arr[arr.length - i - 1];
24      }
25      return newArray;
26    }
27
```

2. **Second Error:** Regarding the second error, the arguments you passed inside the testReverse2 were elements of a double. In line 25, Although you have changed the error in the for loop to assign the new elements of the newly created that is named as `newArray` to hold elements from the condition embedded for the input array `arr`, you did not update the return statement outside of the for loop. Instead of `return arr` in line 25, you should change it to `return newArray` as it is the newly created int[] that holds the elements in the order that the method has intended to do.



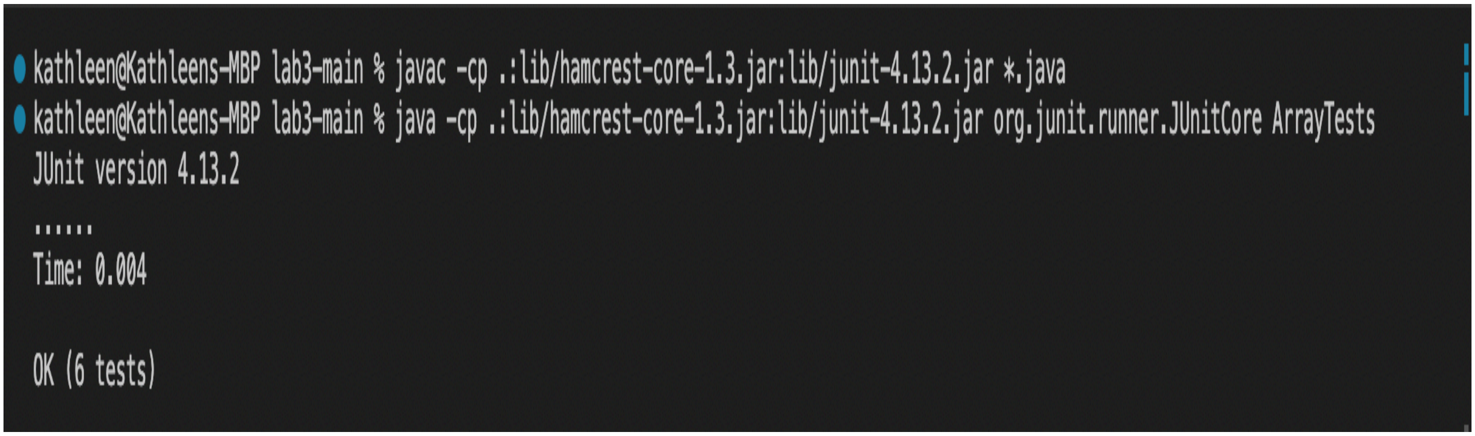
```
16
17    // Returns a *new* array with all the elements of the input array in reversed
18    // order
19    static int[] reversed(int[] arr) {
20      int[] newArray = new int[arr.length];
21      for(int i = 0; i < arr.length; i += 1) {
22        //arr[i] = newArray[arr.length - i - 1];
23        newArray[i] = arr[arr.length - i - 1];
24      }
25      return newArray;
26    }
```

After Debugging

After compiling the tests again using JUnit commands:

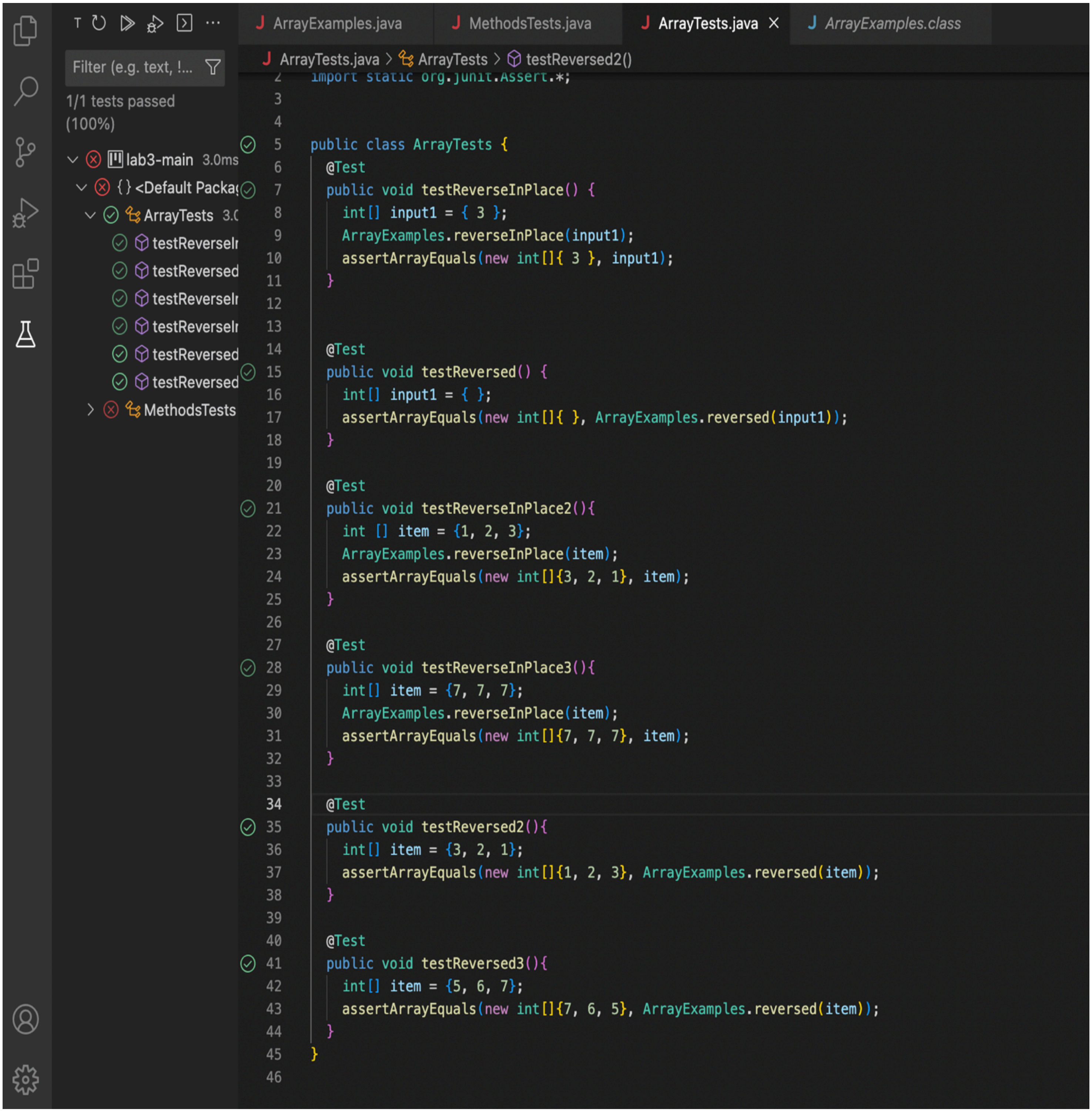
```
$ javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
$ java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUn
```

the tests have all passed as shown in the terminal output.



```
● kathleen@Kathleens-MBP lab3-main % javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
● kathleen@Kathleens-MBP lab3-main % java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUnitCore ArrayTests
  JUnit version 4.13.2
  ......
  Time: 0.004

  OK (6 tests)
```

The JUnit tests have passed within the code as well, with the errors now being solved.



```java
import static org.junit.Assert.*;


public class ArrayTests {
  @Test
  public void testReverseInPlace() {
    int[] input1 = { 3 };
    ArrayExamples.reverseInPlace(input1);
    assertArrayEquals(new int[]{ 3 }, input1);
  }


  @Test
  public void testReversed() {
    int[] input1 = { };
    assertArrayEquals(new int[]{ }, ArrayExamples.reversed(input1));
  }

  @Test
  public void testReverseInPlace2(){
    int [] item = {1, 2, 3};
    ArrayExamples.reverseInPlace(item);
    assertArrayEquals(new int[]{3, 2, 1}, item);
  }

  @Test
  public void testReverseInPlace3(){
    int[] item = {7, 7, 7};
    ArrayExamples.reverseInPlace(item);
    assertArrayEquals(new int[]{7, 7, 7}, item);
  }

  @Test
  public void testReversed2(){
    int[] item = {3, 2, 1};
    assertArrayEquals(new int[]{1, 2, 3}, ArrayExamples.reversed(item));
  }

  @Test
  public void testReversed3(){
    int[] item = {5, 6, 7};
    assertArrayEquals(new int[]{7, 6, 5}, ArrayExamples.reversed(item));
  }
}
```

Information For SetUp:

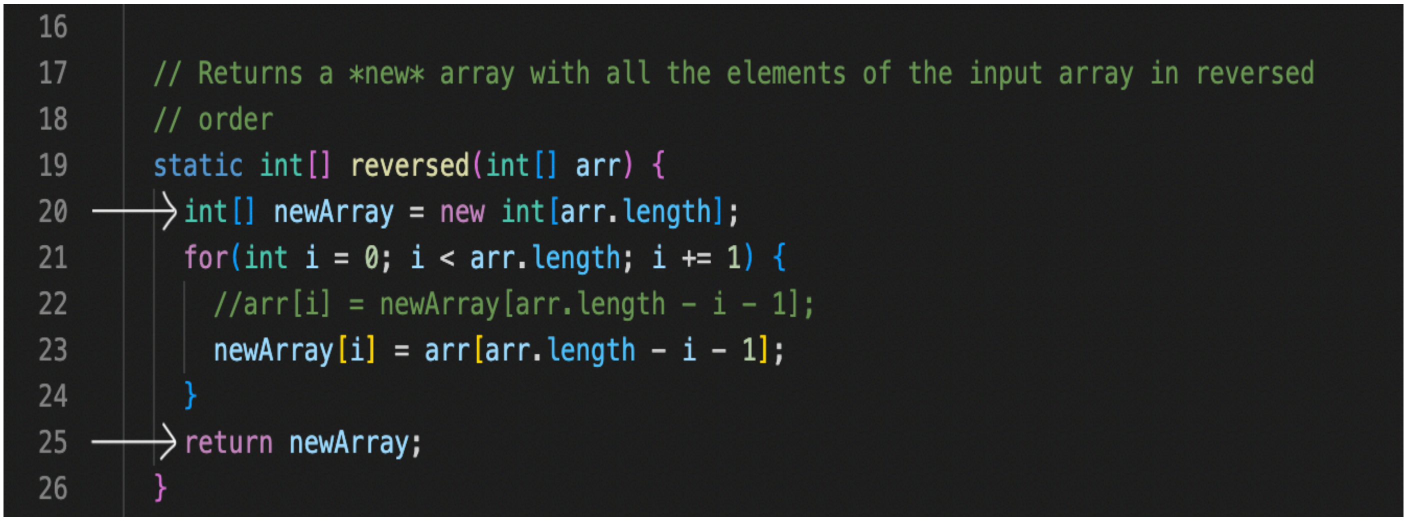
File/Directory:

- The directory that was used is from `/GitHub/lab3`

- Two java files that will help approach this problem is ArrayExamples.java and ArrayTests.java for the setup.
  - `ArrayExamples.java` : This file has the methods that was called when testing in the ArrayTests.java, specifically reverseing methods.
  - `ArrayTests.java` : This file has all the testers, implementing different testing scenarios to see if the reverse methods from ArrayExamples.java induce the correct expected output as intended.

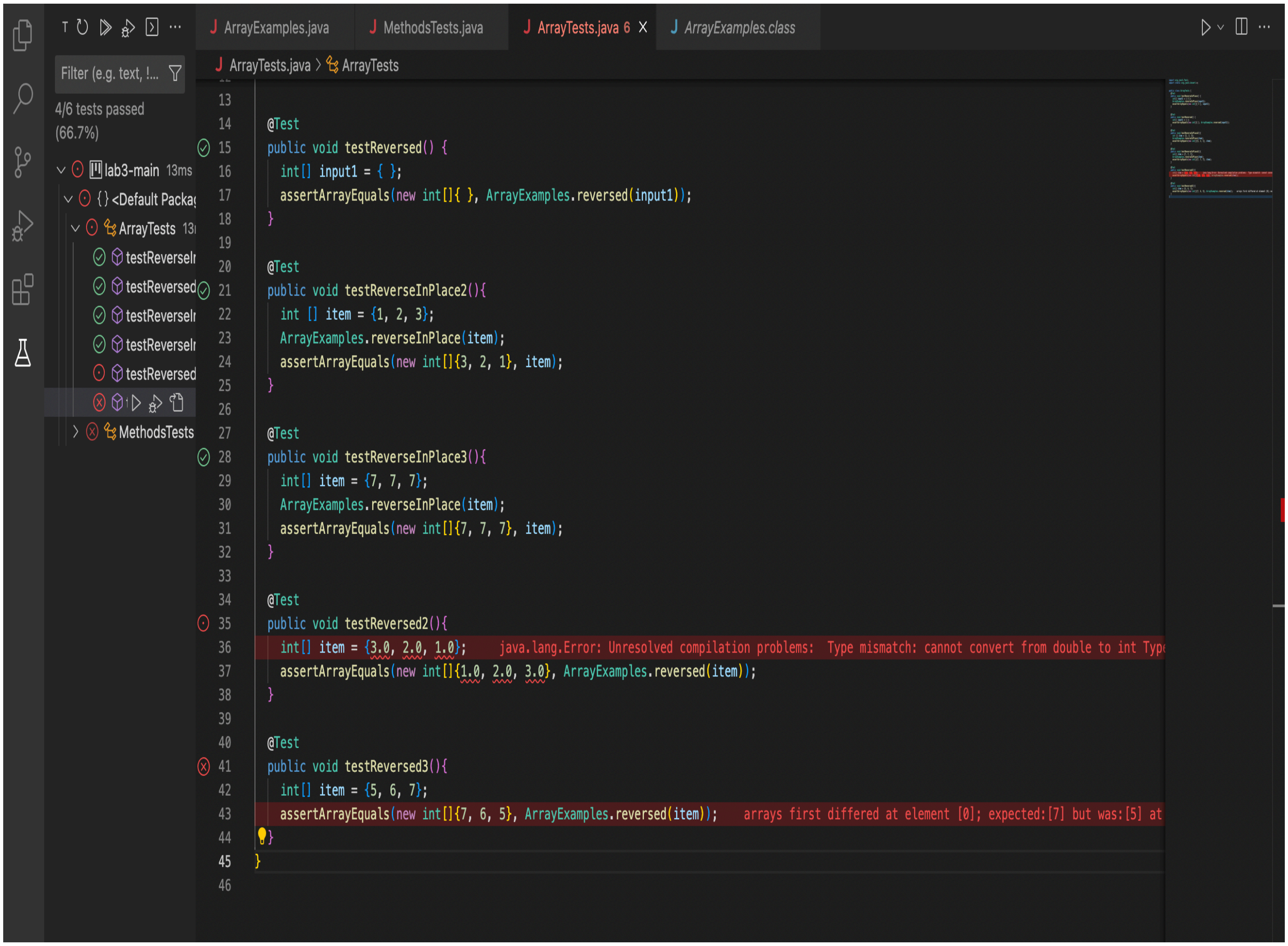
Related Contents of Errors:

- ArrayExamples.java file containing reversed method to debug



```

// Returns a *new* array with all the elements of the input array in reversed
// order
static int[] reversed(int[] arr) {
  int[] newArray = new int[arr.length];
  for(int i = 0; i < arr.length; i += 1) {
    //arr[i] = newArray[arr.length - i - 1];
    newArray[i] = arr[arr.length - i - 1];
  }
  return newArray;
}
```

- Errors in the specific tests within ArrayTests.java



```
13
14    @Test
15    public void testReversed() {
16      int[] input1 = { };
17      assertArrayEquals(new int[]{ }, ArrayExamples.reversed(input1));
18    }
19
20    @Test
21    public void testReverseInPlace2(){
22      int [] item = {1, 2, 3};
23      ArrayExamples.reverseInPlace(item);
24      assertArrayEquals(new int[]{3, 2, 1}, item);
25    }
26
27    @Test
28    public void testReverseInPlace3(){
29      int[] item = {7, 7, 7};
30      ArrayExamples.reverseInPlace(item);
31      assertArrayEquals(new int[]{7, 7, 7}, item);
32    }
33
34    @Test
35    public void testReversed2(){
36      int[] item = {3.0, 2.0, 1.0};     java.lang.Error: Unresolved compilation problems:  Type mismatch: cannot convert from double to int Typ
37      assertArrayEquals(new int[]{1.0, 2.0, 3.0}, ArrayExamples.reversed(item));
38    }
39
40    @Test
41    public void testReversed3(){
42      int[] item = {5, 6, 7};
43      assertArrayEquals(new int[]{7, 6, 5}, ArrayExamples.reversed(item));    arrays first differed at element [0]; expected:[7] but was:[5] at
44    }
45  }
46
```

**Compile/Commands

- Compiling the files with JUnit commands

  ```
  $ javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
  ```

  which then produces the errors in the tests (failed).

Approach to the Problem

1. The argument being passed in for testReversed2 within ArrayTests.java should be integer type instead of double type in lines 35 and 36. Also, the reversed method in ArrayExamples.java should return newArray in line 25.

   - **Error:**

1st Problem: Type Mismatch
Line 35

```
int[] item = {3.0, 2.0, 1.0};
```

Line 36

```
assertArrayEquals(new int[]{1.0, 2.0, 3.0}, ArrayExamples.reversed(item
```

2nd Problem: Tests Failed Line 25

```
return arr;
```

- **Resolve:**

-[✔] Solution to 1st Problem:
Line 35

```
int[] item = {3, 2, 1};
```

Line 36

```
assertArrayEquals(new int[]{1, 2, 3}, ArrayExamples.reversed(item));
```

-[✔] Solution to 2nd Problem:
Line 25

```
return newArray;
```

Reflection

Something that have always been helpful for me was learning the different shortcut linux commands. It is very useful when I code as I can easily see where I am at and how I can see contents of a specific file just by inputting certain commands into the terminal. For example in this lab assignment, I was not able to run any of my files for the lab 3 folder. However, I used what I learned in class and did:

1. cd
2. ls
3. cd Downloads
4. cd lab3-main
5. ls

Then after these commands, I was able to run my ArrayTests.java. This is a longer approach due to the reason that I download the zip file from GitHub instead of using the existing file from GitHubt deskstop that I already have created and altered for lab 3’s week.

This inspired me to learn more commands because it is very time efficient when I can just do things straight from the terminal instead of closing and opening tabs and doing a lot of clicking with my track pad. Several of very useful commands that I have used are:

- mv command
  - This command allows you to rename a file.
  - Correct syntax: `mv old_filename.txt new_filename.txt`
- locate command
  - This command allows you to find any file in the database. By using -i after the command will help with case sensitivity.
  - Correct syntax: `locate -i to_Find` if you are searching with one word, `locate -i to_Find*to_Find` if you are searching using two words
- du command
  - This command allows you to see the amount of space a file or a directory takes up.
  - Correct syntax: `du /home/user/Documents`
  - Using -m will provide the information for the folder and file in MB

  - Using k will display the information in KB
  - Using -h will show the last time stamp the folders and files were modified