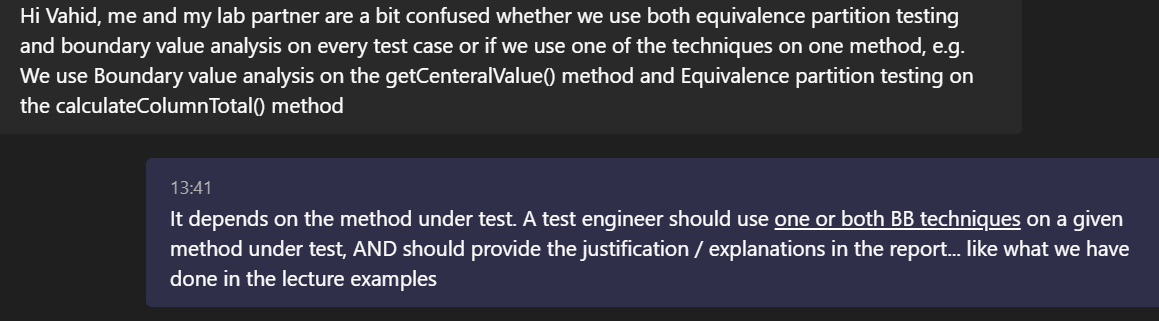
Notes:

* This is a “LIVE“ document, i.e., more examples and items will be added to this document as we progress in the labs.
* Getting issues/error messages in the project set-up are often due to not following the lab document step-by-step carefully and/or not installing Java/JDL and the Eclipse IDE properly
* Each given issue/error message in the project set-up could be due to MANY possible root causes. We discuss the solution ideas that we are aware of. Students are encouraged to review online resources (such as StackOverflow) also which offer many other solution ideas as well.

# Technical FAQ (about testing)

## Using Black-box test-case design techniques



## Can the Black-box test-case design examples, solved in the class, be used as one of the five methods under test in the Lab2 doc?

**NO.** Those examples can be used as examples for students to learn, but they CANNOT be included as one of the five methods under test, as asked by the Lab2 doc.

You should design test-cases and then develop JUnit test methods, for SUT methods which have NOT been solved in class examples.

## Developing JUnit test code versus “design” of test cases

*“Hi Vahid, I have wrote code for the 5 chosen methods from the Range class in Eclipse, but I don't understand how I am supposed to document them as test cases in the report?”*

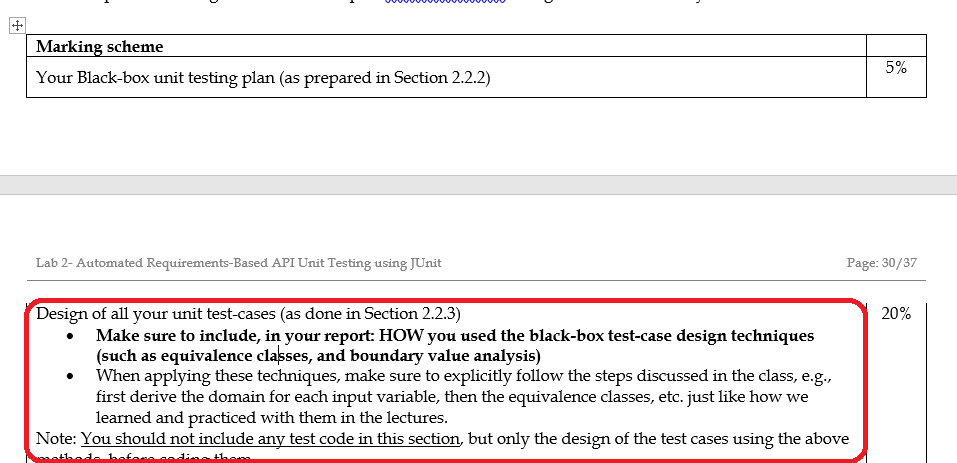
**Answer:**

*RE: I have wrote code for the 5 chosen methods from the Range class*

Do you mean you have developed the JUnit "test" code for those 5 methods?

*RE: I don't understand how I am supposed to document them as test cases in the report?*

A section of the report shall include the design of all your unit test-cases



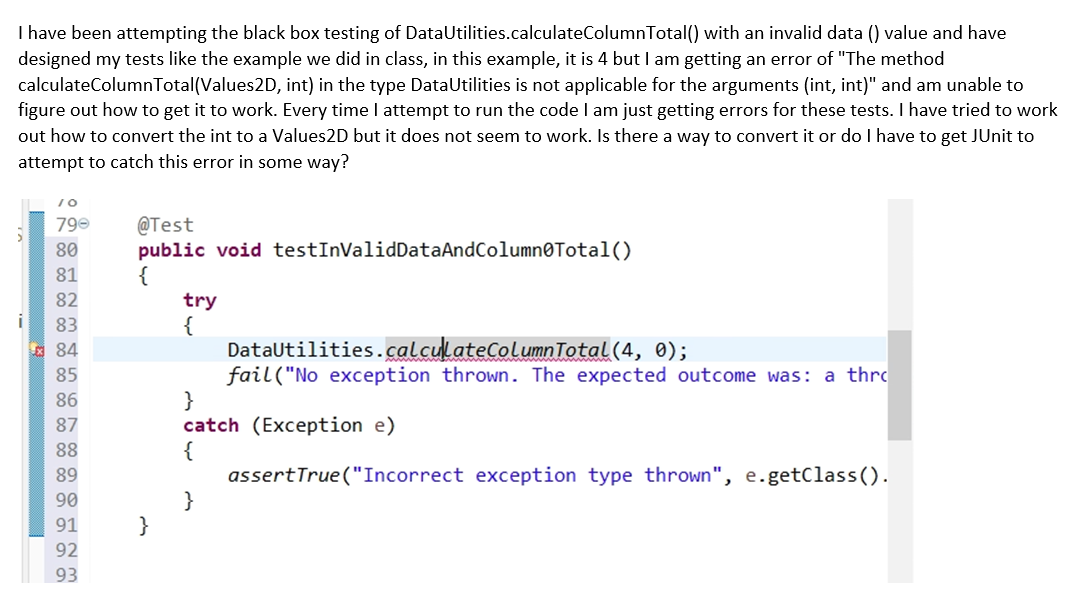
In lectures, we have solved many examples of test-case design. Pls refer to them

## How many test cases should we do for each method under test?

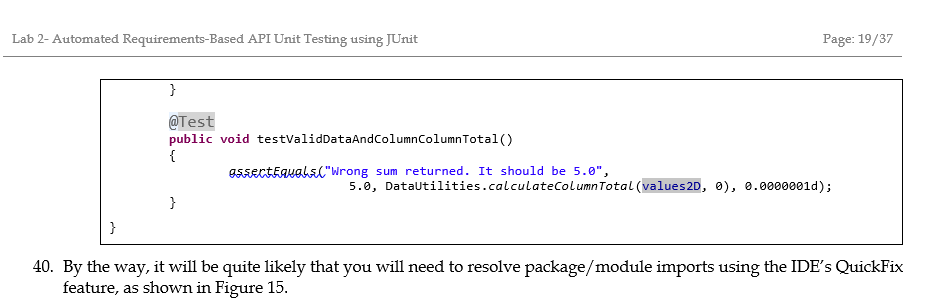
As the lab doc says, students shall apply the “Strong” equivalence-class partitioning (multi-dimensional) and boundary value analysis (BVA) techniques on the methods under tests, as appropriate.

As we have learned in the lectures, these test-design techniques will help us design a set of test cases for a given method under test. Thus, we cannot reveal the number of test cases for each method. But that number would be determined by “correct” application of those test-design techniques, similarly as done in many examples in the lectures.

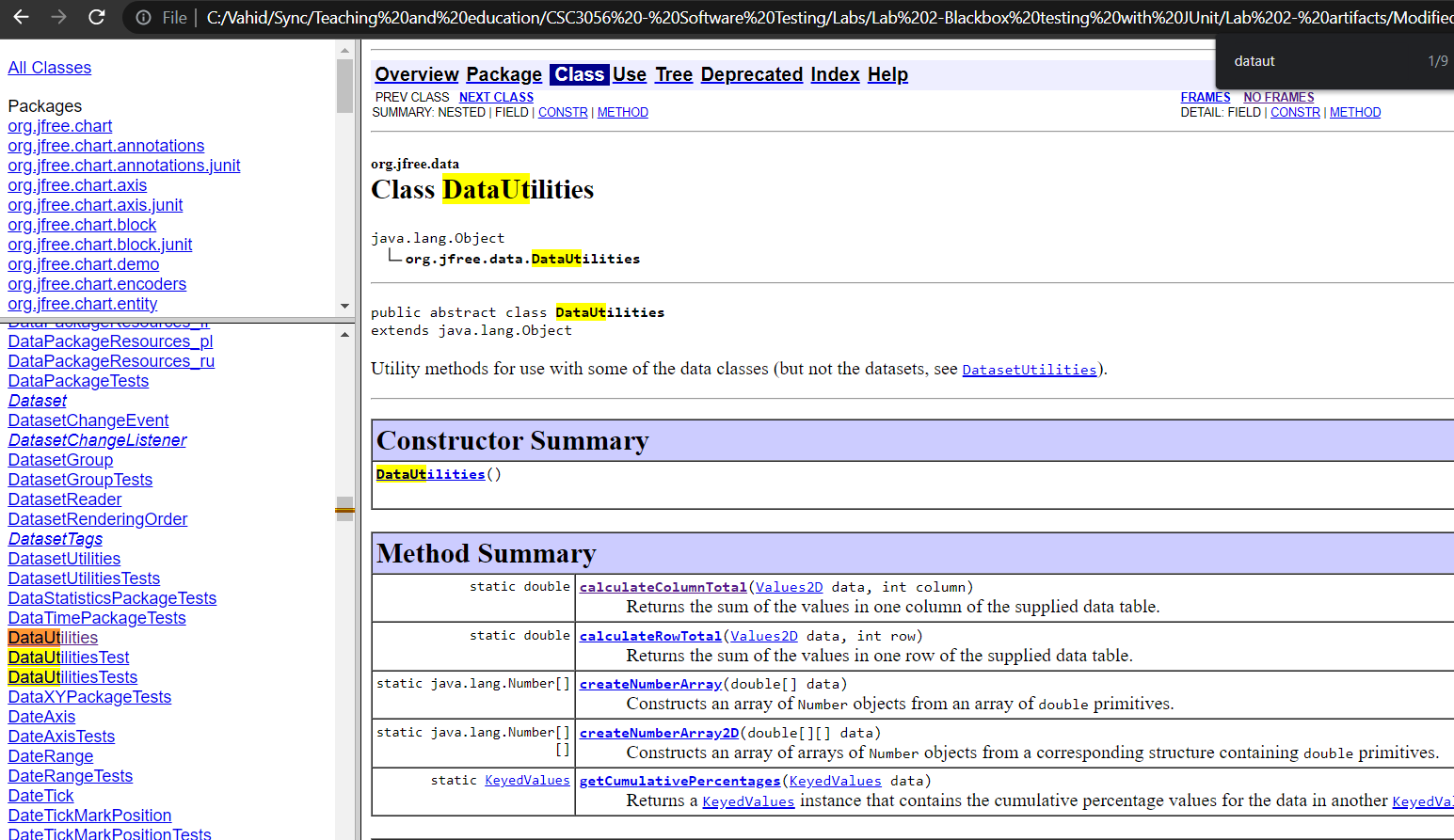
## Proper calling of the methods under test



As we see in the lab2 doc example, calculateColumnTotal needs a matrix for its first input:

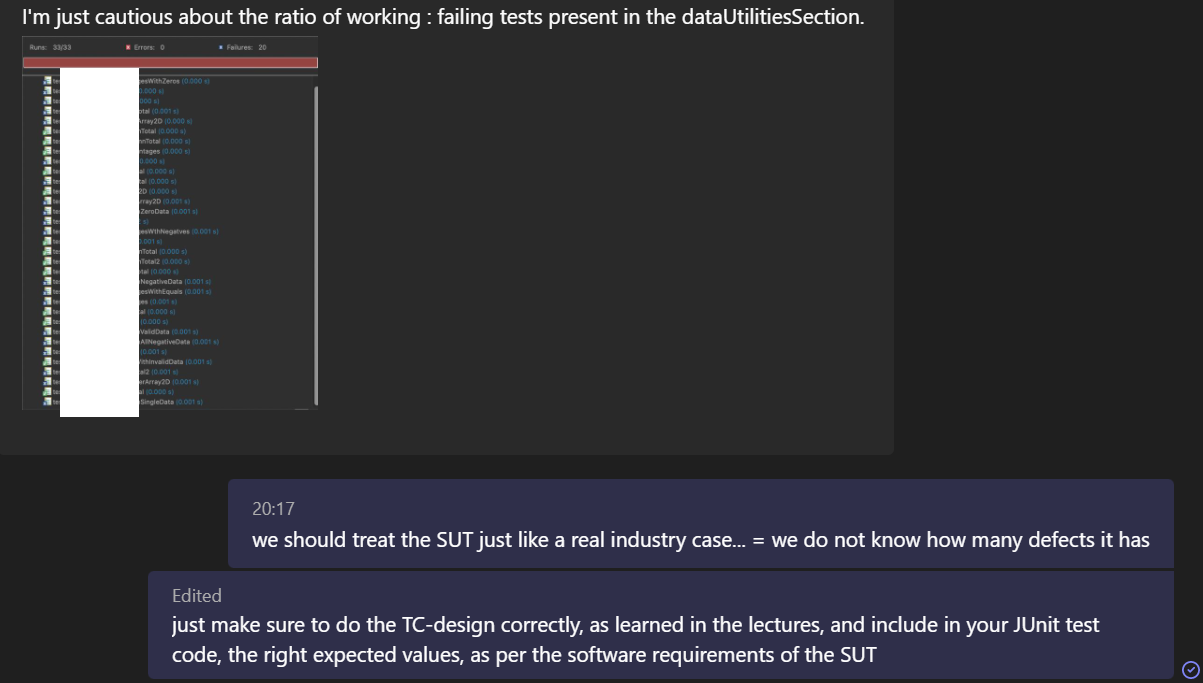


JavaDoc also specifies that:



Thus, we cannot call the method with arguments (int, int).

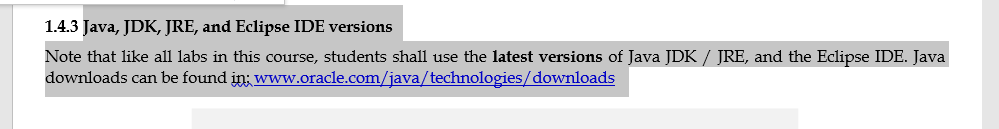
## Number of defects in the SUT, and the expected number of test-case failures



# Non-technical FAQ (Project set-up, IDE, etc.)

## *Need to install both JDK and JRE*

Important note for Lab2, from the lab doc:



In addition to installing JDK and JRE on your machine, you need to ensure they are set up properly inside your Eclipse settings and also Project settings. See these help pages:

* <https://www.google.com/search?q=setup+jdk+jre+in+eclipse>
* <https://wiki.eclipse.org/Eclipse/Installation>

## Getting the error message "Could not find or load main class” in Java / Eclipse

Your project set up in the IDE has issues, or you need to use Eclipse in the right way. ... See:

<https://www.google.com/search?q=eclipse+java+could+not+find+or+load+main+class>

## “I was wondering if we could use IntelliJ for the lab assignment?”

The labs have been designed using the Eclipse IDE. If a student wants to use another IDE, it is up to them to port the settings and steps, etc.; and we will be unable to provide tool support.

## *“No JUnit tests found.*”

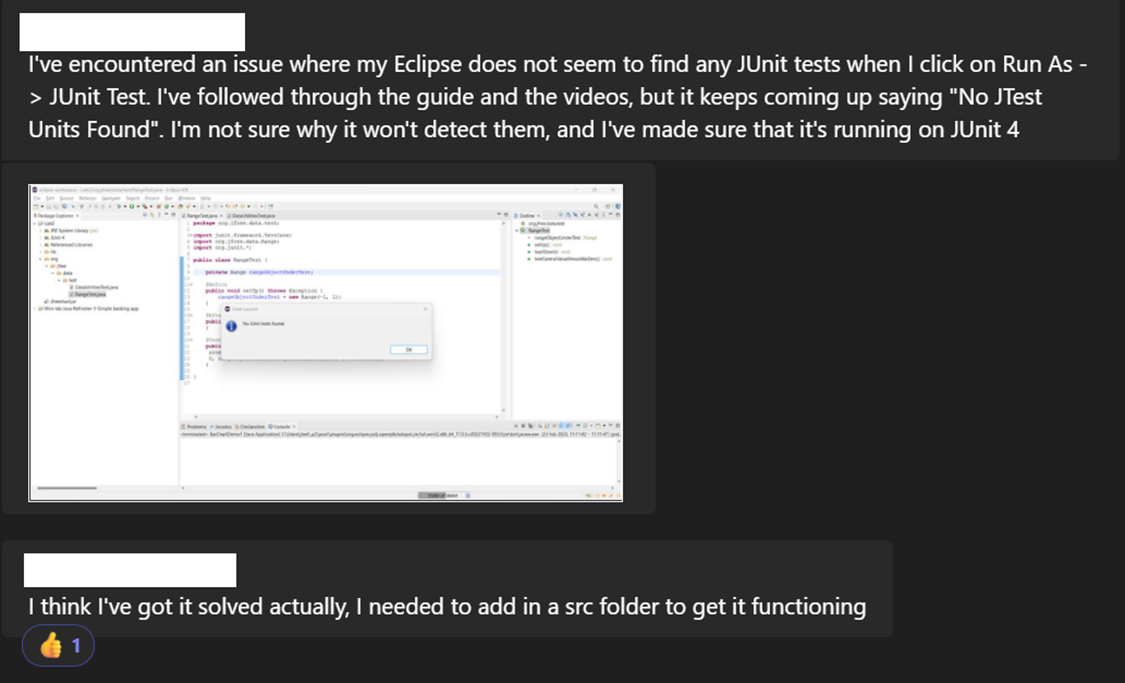
If the student/group follows the lab document step-by-step carefully, this message should not occur.

Among many possible root causes, this could be due to not properly setting up your Eclipse project to use the JUnit version 4. Lab2 doc has a subsection and even a video on how to set up your project to use JUnit4.

Online resources (such as StackOverflow) have also many other suggestions:

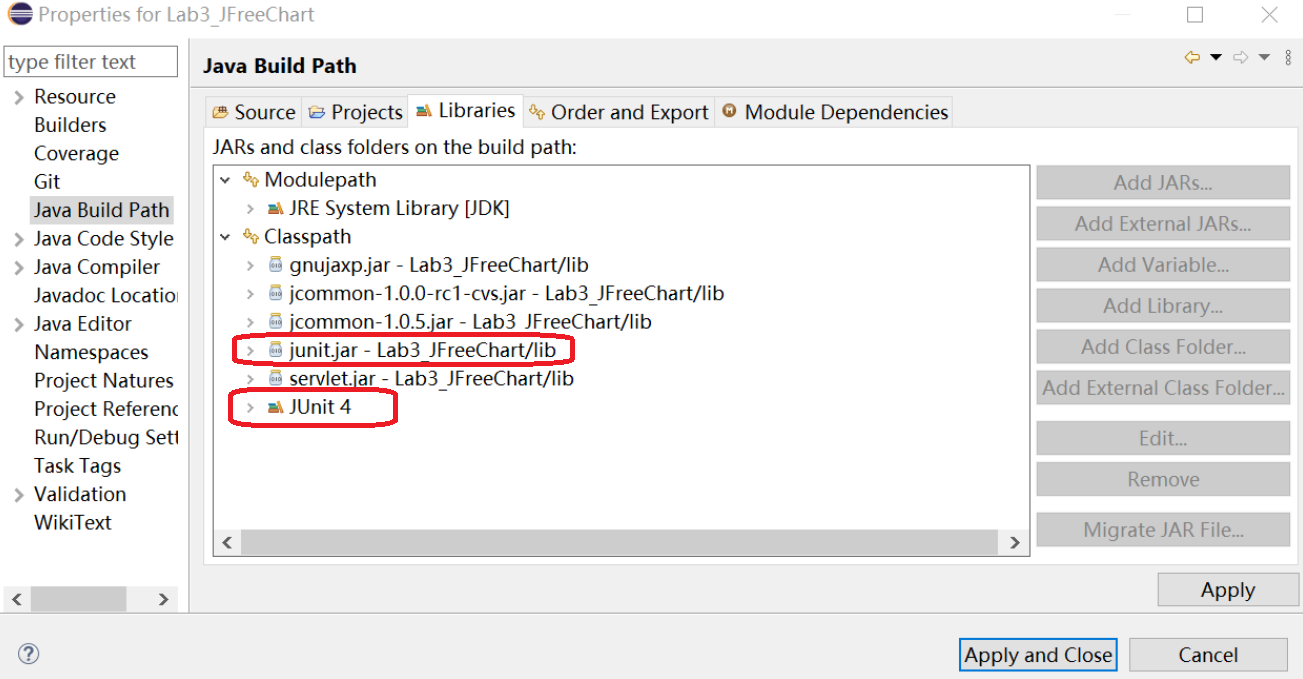
<https://www.google.com/search?q=no+junit+tests+found+eclipse>

Also, a fix done by a group:



## My test-code is mixing up JUnit versions

My test-code is mixing up JUnit versions. Here is my project settings:

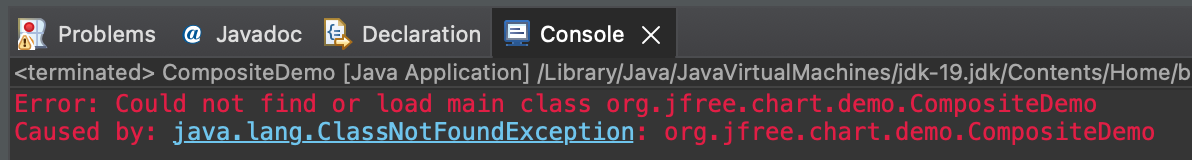


**Answer:**

There is a clear mistake above, as **both** JUnit4 and also junit.jar (that is for JUnit 3) have been included. You cannot include both JUnit versions!! And thus you should NOT include the junit.jar file, but only include JUnit4 in your project. You should just delete junit.jar from the list above.

## ClassNotFoundException

If the student/group follows the lab document step-by-step carefully, this message should not occur.

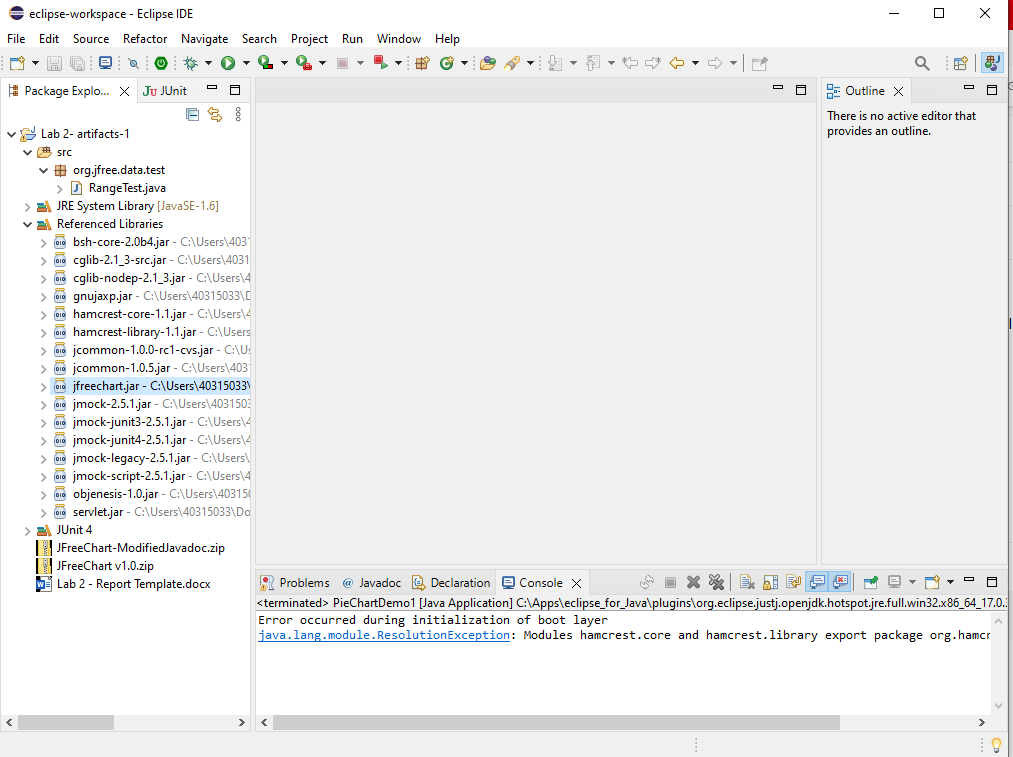


Lots of advice online: <https://www.google.com/search?q=java+clasnotfoundexception>

Some hints:

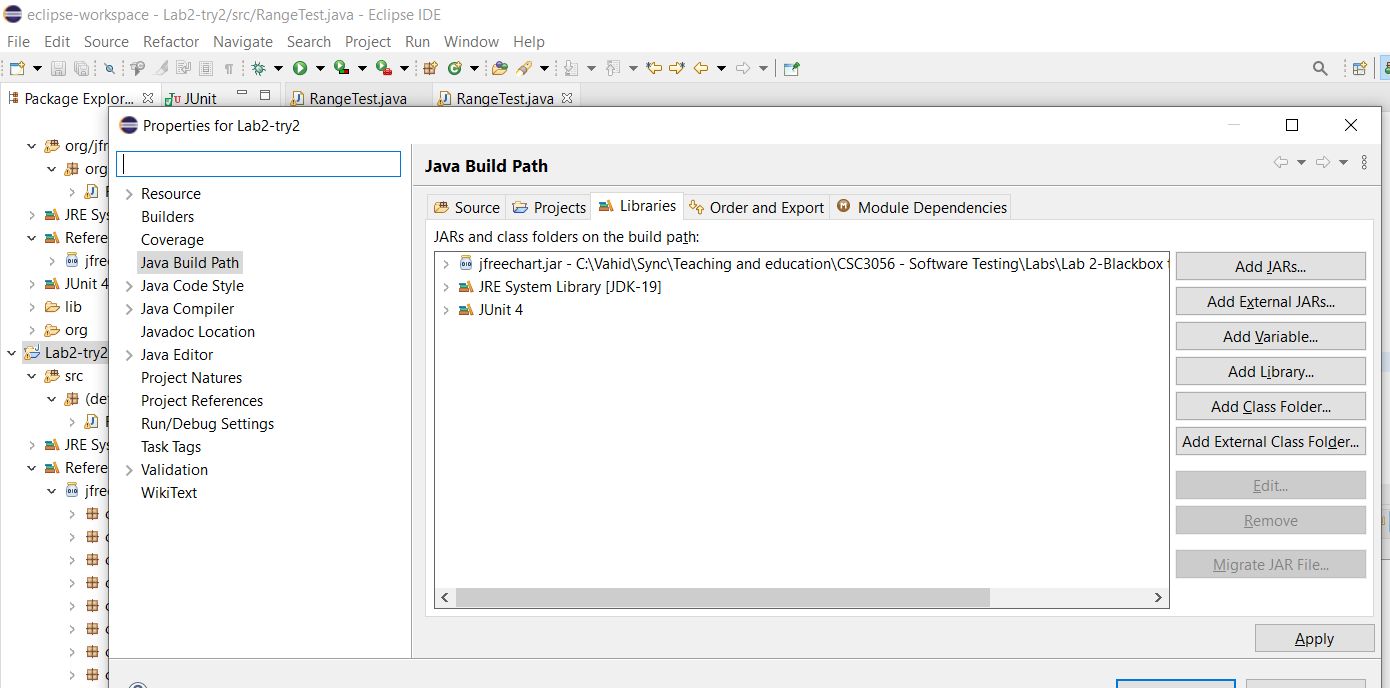
* Find out which JAR file contains the problematic Java class.
* Check whether this JAR is present in the application classpath.
* If that JAR is already present in the classpath, make sure the classpath is not overridden (e.g. by a start-up script).
* …

## Error during initialization of boot layer



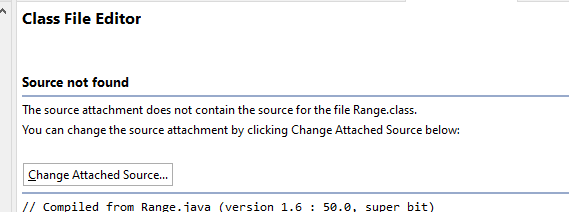
Suggested solution:

* Go to the *Libraries tab* of the *Project Properties* window
* Remove all but the following three items from the list
* Try running the demos of JFreeChart again



## Cannot edit .CLASS files

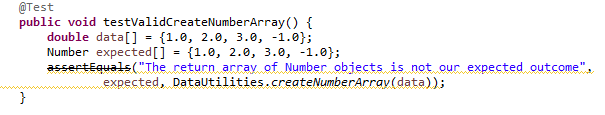
Issue: *“When we are trying to view the classes in eclipse we are getting this error. We have tried to attach the jfreechart.jar file to it but it does not change anything. Have you got any idea what's wrong with it? We have tried it on two different computers.”*



**Answer:**

You should open the .JAVA source files. .CLASS files are the compiled “binary” files and cannot be viewed / changed by humans.

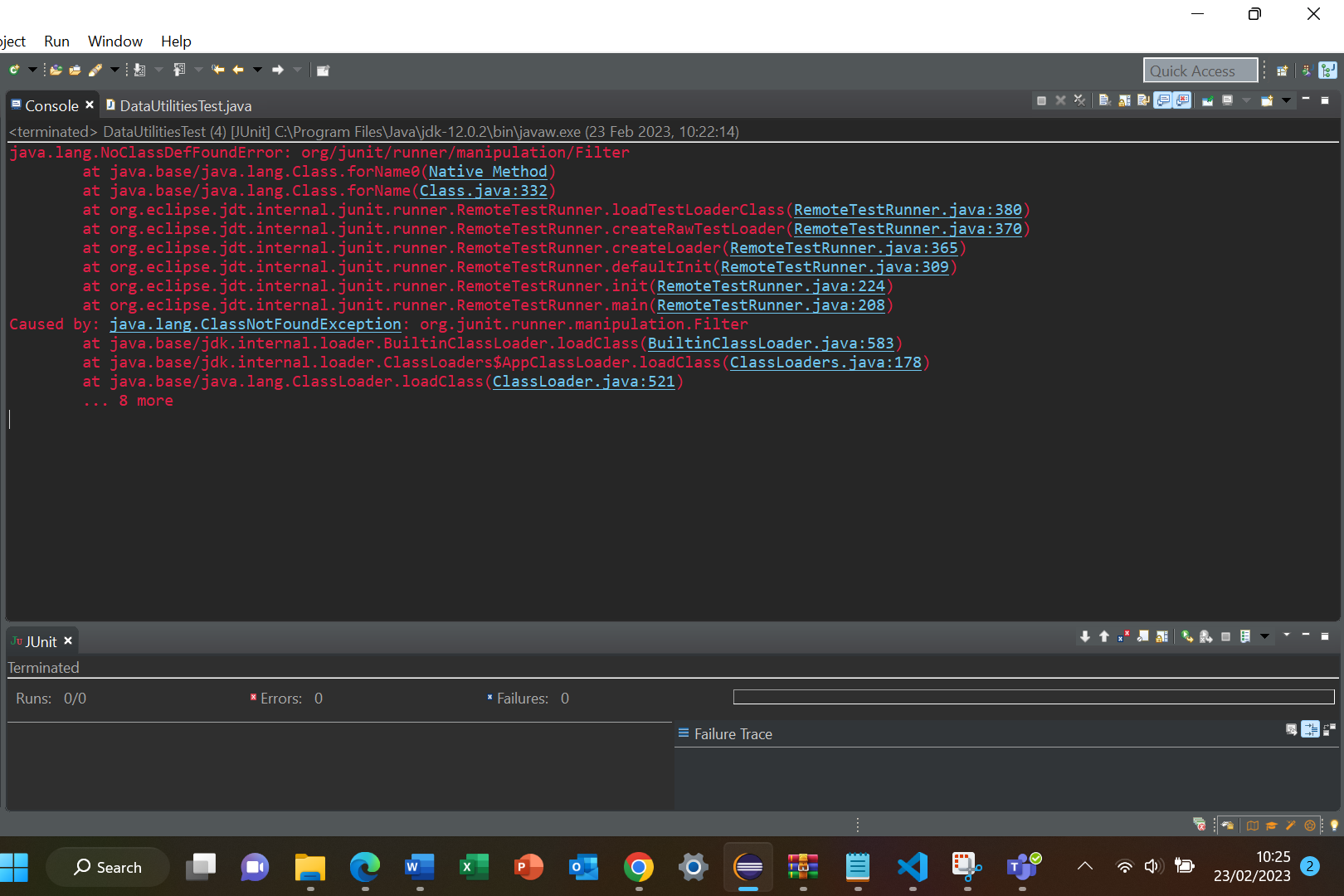
## The IDE says: assertEquals is deprecated



It could be that: *You're using junit.framework.Assert* instead of *org.junit.Assert.*

See:  [*https://stackoverflow.com/questions/22541455/warning-the-method-assertequals-from-the-type-assert-is-deprecated*](https://stackoverflow.com/questions/22541455/warning-the-method-assertequals-from-the-type-assert-is-deprecated)

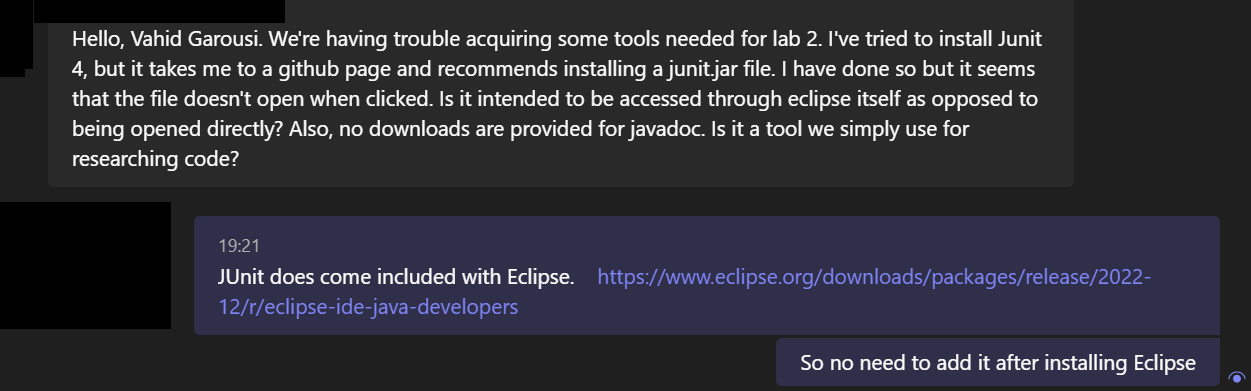
## NoClassDefFoundError and error in RemoteTestRunner



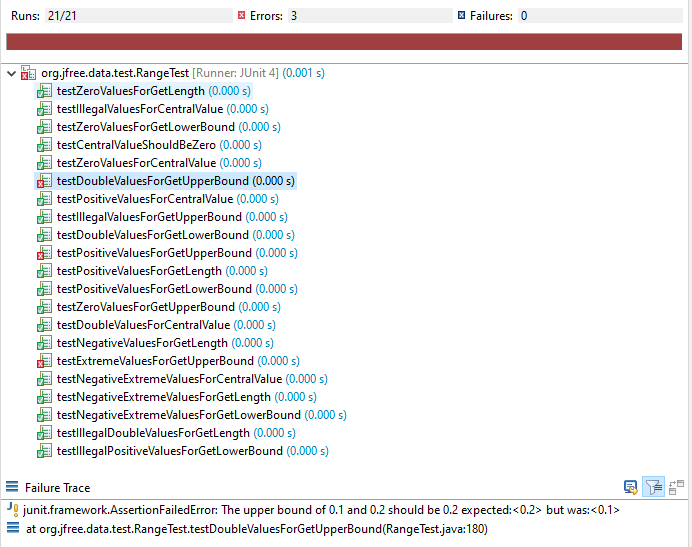
Among many possible root causes, this could be due to not properly setting up your Eclipse project to use the JUnit version 4. Lab2 doc has a subsection and even a video on how to set up your project to use JUnit4.

See more details here:<https://stackoverflow.com/questions/33521436/junit-noclassdeffounderror-org-junit-runner-manipulation-filter>

## Shall we install JUnit separately on Eclipse?

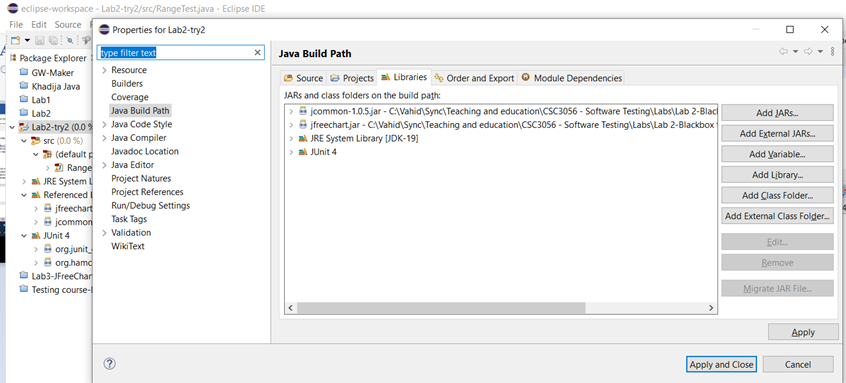


## Getting AssertionFailedError

*When running tests for the getUpperBound method in the Range class, errors are occurring, however I believe they should be failures based on the stack trace (see below).*  

One root cause can be that your project is using JUnit versions < 4. See the lab doc on how to set your project to use JUnit. Also, see: <https://www.google.com/search?q=AssertionFailedError>

Also make sure not do NOT have the older JUnit version JAR’s in the included libraries of your project:



assertionFailedError is a JUnit exception, thrown when an assertion fails.

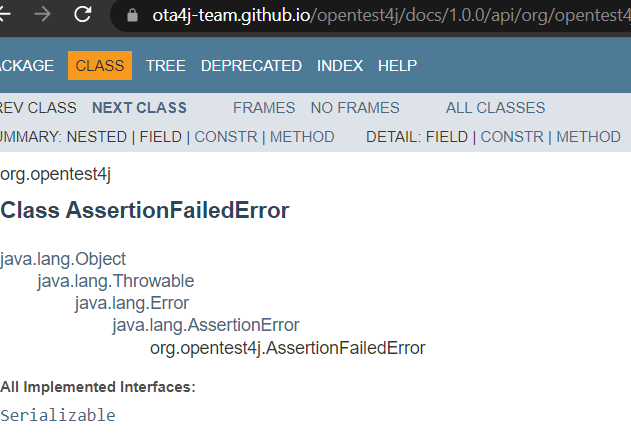
When we do our project setup and JUnit library imports properly, assertionFailedError should be shown as "Failure" in JUnit output window, and NOT as an "Error".

Things would get mixed up, if you have set up your project to use the JUnit version, incompatible with your test-code use of JUnit framework. JUnit versions 3 and 4 treat the assertionFailedError differently.

If assertionFailedError is treated as a failure in Eclipse JUnit output view of your project, it would be OK, and would mean that you are using the right JUnit version (v4).

For more info about assertionFailedError, see:

* Class hierarchy of assertionFailedError



From [https://ota4j-team.github.io/opentest4j/docs/1.0.0/api/org/opentest4j/AssertionFailedError.html](https://ota4j-team.github.io/opentest4j/docs/1.0.0/api/org/opentest4j/AssertionFailedError.html?is-external=true)

* <https://www.google.com/search?q=AssertionFailedError>