**CS 302 Data Structures Project 1**

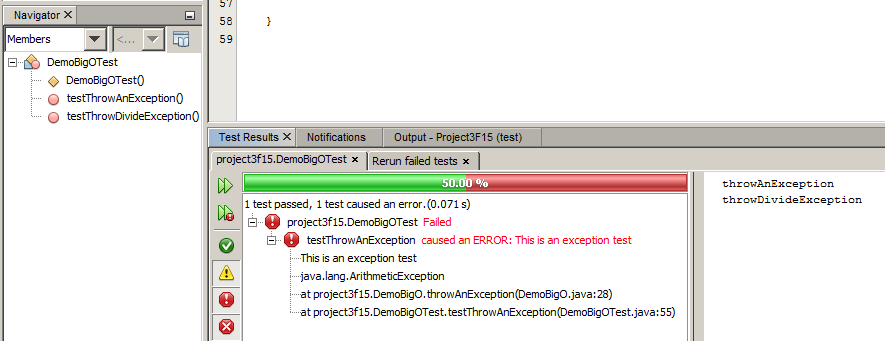
**Objective**

This assignment will help you get acquainted with O(n), more testing, and begin exception handling.

**Program assignment**

Create the jUnit test class and write two jUnit test methods. Only one test method will work as the second method just divides by zero and aborts. Fill in the methods for O(1), O(n), and O(n\*n). The methods with O(log n) and O(n log n) are done. Include the three output files which are generated separately.

**Design specifications**

* Use the given main method Project1 class.
* Create a Java project Project1 and a package project1. The class Project1 will have a main method and the method called “run”. The main method will only call method “run”. The method called “run” will call all of the methods in the class DemoBigO described below. Copy and paste my code into your project.
* Use the demo and driver of the given main and class DemoBigO. Copy and paste my code into your project.
* In DemoBigO class create methods which contain and computationally verify algorithms of a) O(1), b) O(log n), c) O(n), d) O( n log(n) ), e) O(n**2**). Parts b and d are done for you: That means use them as they are with no modifications!
* Also in the DemoBigO class create a method that throws an exception by using the throw java key word. The other method dot **not** use the throw java key word but generates the exception by attempting to actually divide by zero at run tim..
* Add a JUnit test class and method and use the default name DemoBigOTest. The method that is automatically added is named after the identifier that you used in the method above. Write the test methods testThrowDivideException and testThrowAnException. No other methods will need to be tested. Put a screenshot of the successful test in your Project2 folder. 

Why for the test output is there a 50% pass, why don’t both tests fail?

Method throwDivideException() divides one/zero at run time and the compiler throws the exception and is handled in jUnit. It returns to testThrowDivideException() and assertEquals() says it passed.

Method throwAnException() is coded to throw an exception. That’s all it does. So throwAnException() is void as a return is unreachable. Then testThrowAnException() can create a DemoBigO object and invoke instance.throwAnException() but there can’t be a return, you can’t use an assertEquals() as there is nothing to compare in my version.

I suspect that it is possible to write code in a way that it meets the project specifications **and** has both test past or neither test pass.

I have seen students write tests for the other methods. That is not required, but is good practice. The graders are not allowed to grade off for this even though their test output looks different than mine.

Write the method “throwAnException” in “DemoBigO” class and call it from the run method given as

// demo throwing exception

// comment it out after using it as it throws an exception and stops

DemoBigO objThrow = new DemoBigO();

objThrow.throwAnException();

that produces output

Exception in thread "main" java.lang.ArithmeticException: This is an exception test

at project1.DemoBigO.throwAnException(DemoBigO.java:26)

at project1.Project1.run(Project1.java:27)

at project1.Project1.main(Project1.java:14)

Java Result: 1

**Then comment it out to produce the other outputs.** Or all you will get is that crash report!

// demo throwing exception

// comment out when not using – it throws an exception and stops

/\*DemoBigO objThrow = new DemoBigO();

objThrow.throwAnException();\*/

* Finish the demoBigO1 for O(1) method. You can verify that it is working from obtaining output ***similar*** to my output.
* Finish the “demoBigOn” for O(n) method.

Finish the “demoBigOnn” for O(n\*n) method.

**What to turn in to your TA**

Zip the project and upload it to Canvas. It must include your three source code classes, Project3, DemoBigO, and DemoBigOTest. It will include your test source code, screenshots or copies of the 3 output files: “Output WITH deliberate throw of an exception”, “Output without the deliberate throw of an exception”(Your Big O output, and “Test Output without the deliberate throw of an exception” to Canvas Project3 assignment called P3 in Canvas.

**Grading criteria ( 8 points maximum )**

1. Test method DemoBigOTest class created and working …..………………..……... 1 pt
2. DemoBigO class ( Mostly given, fill in steps 3-7) ……………………………..…... 1
3. Completed methods throwDivideException, throwAnException …..…………...… 1
4. Completed method demoBigO1 ……..….………………………………………..… 1
5. Completed method demoBigOn ………………………………………………….... 1
6. Completed method demoBigOnn …………………………………………………... 1
7. Output of throwing an exception; Output of causing a divide by zero exception; Output from Big O test ………………………………….…………………………………………………… 1
8. Naming Conventions, Style and Comments, Coding Efficiency ….….1

Project 1 Output without the deliberate throw of an exception

run:

O(1) method calculational verification

Trial 1

n= 1 time/1= 1399.0

n= 10 time/1= 350.0

n= 100 time/1= 350.0

n= 1000 time/1= 0.0

n= 10000 time/1= 0.0

n= 100000 time/1= 350.0

n= 1000000 time/1= 350.0

n= 10000000 time/1= 0.0

Since time/1 is not growing with n demoBigO1 is O(1)

……..

O(n\*n) method calculational verification

n= 1 time/n\*n= 8748.0

n= 10 time/n\*n= 27.99

n= 100 time/n\*n= 24.2508

n= 1000 time/n\*n= 3.70516

n= 10000 time/n\*n= 0.94031571

n= 100000 time/n\*n= 0.9058009488

Since time/n\*n is not growing with n demoBigOnn is O(n\*n)