Lab 6 - Exception Handling Lab

Worksheet 1 (Monday)

1. Explain when exception handling should be used.

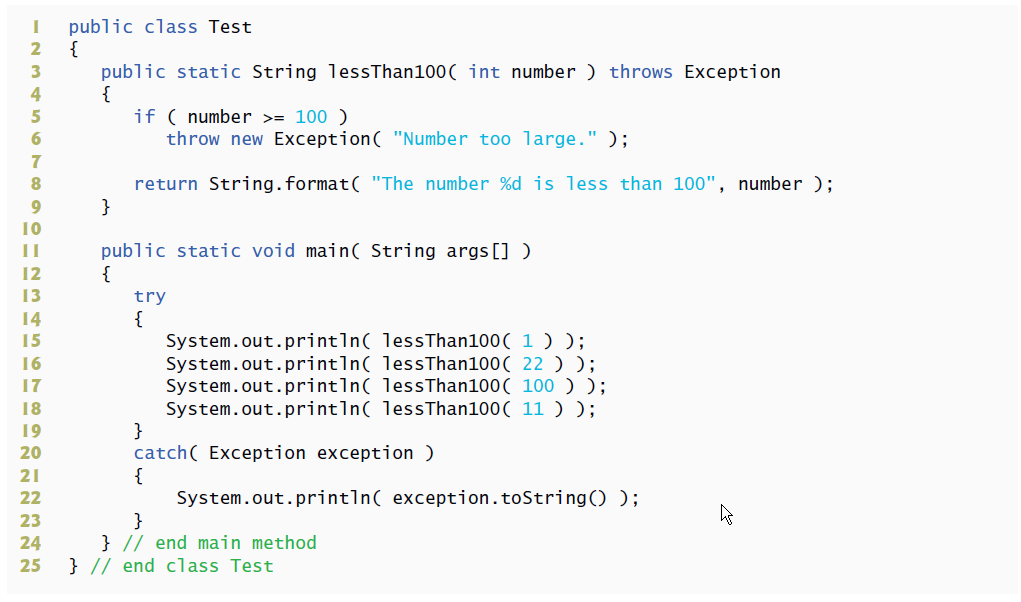
When dealing with synchronous errors and If you can do something to mitigate the exception, or make it look better to the user then use exception handling to create a soft error, instead of having the program crash

1. Describe the general flow of control through a try…catch…finally when an exception occurs and is

caught.

The exception Is caught, it goes to the first catch function of that error type, executes that code, and then goes to the finally block.

1. What is output by the following application?



The number 1 is less than 100

The number 22 is less than 100

Number too large

1. What is the output of this program?

The Followin exception occurrred in method 2

Runtime exception occurrred in method 3

The Following exception occurred in Main

Runtime exception occurrred in method 3

The Following exception occurred in Main

Runtime exception occurrred in method 3



1. What is output by the following program if the user enters the values 3 and 4.7?

java.lang.NumberFormatException: For input string: "4.7"



Worksheet 2 (Wednesday)

Correcting the code:

1. The following code segment should catch only NumberFormatExceptions and display an error message dialog if such an exception occurs:



Change the first line to: Catch ( NumberFormatException exception)

There should be brackets after the first line and after the last line

1. In the following code segment, assume that method1 can throw both NumberFormatExceptions and ArithmeticExceptions. The following code segment should provide appropriate exception handlers for each exception type and should display an appropriate error message dialog in each case:

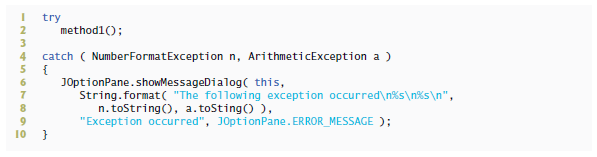
Catch (NumberFormatException n){

jOptionPane.showMessageDialog(null, “The Following NumberFormatException occurred: ” + n.toString() , “NumberFormatException Occurred”, JOptionPane.ERROR\_MESSAGE );

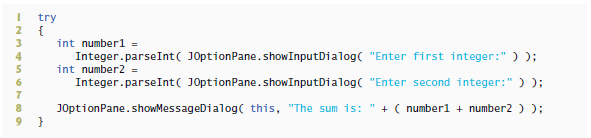
}catch(ArithmaticException a){

jOptionPane.showMessageDialog(null, “The Following Arithmatic Exception occurred: ” + a.toString() , “Arithmatic Exception Occurred”, JOptionPane.ERROR\_MESSAGE );

}



1. The following code segment should display an error message dialog if the user does not enter two integers:



Catch (NumberFormatException n){

jOptionPane.showMessageDialog(null, “Please Enter Only Integers!” , “NumberFormatException Occurred”, JOptionPane.ERROR\_MESSAGE );

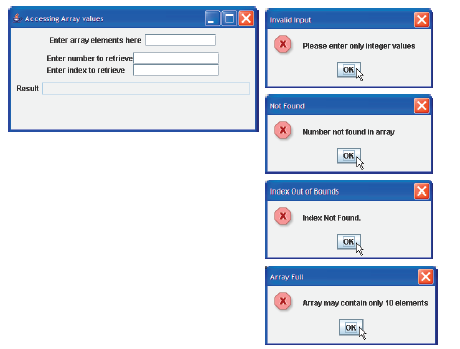
}

Worksheet 3 - Implementation (Wednesday)

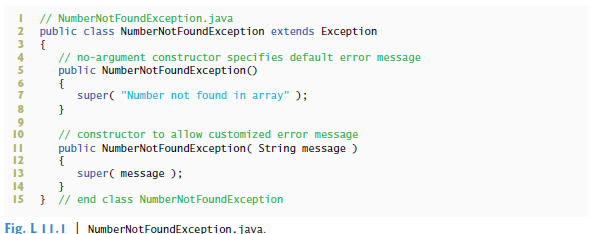
Write a program that allows a user to input integer values into a 10-element array and search the array. The program should allow the user to retrieve values from the array by index or by specifying a value to locate. The program should handle any exceptions that might arise when inputting values or accessing array elements. The program should throw a NumberNotFoundException (Fig. L 11.1) if a particular value cannot be found in the

array during a search. If an attempt is made to access an element outside the array bounds, catch the ArrayIndexOutOfBoundsException and display an appropriate error message. Also, the program should throw an Array-IndexOutOfBoundsException if an attempt is made to access an element for which the user has not yet input a value.

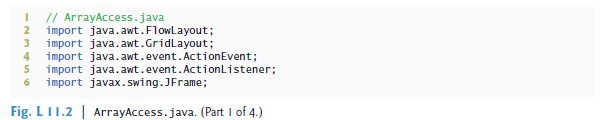
Sample Output:

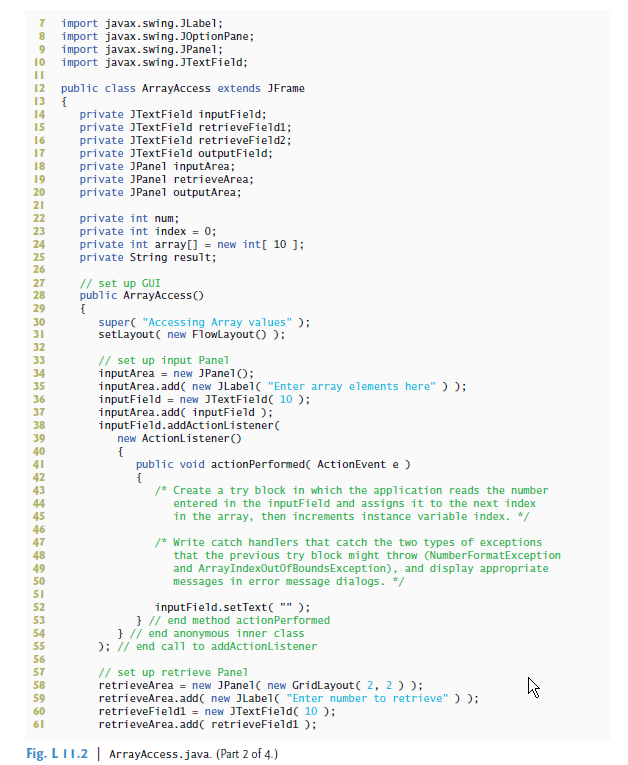


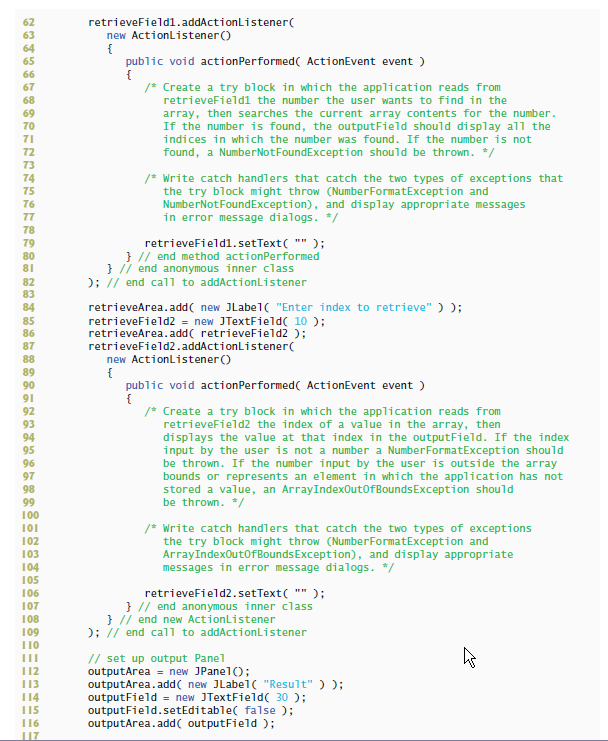
Step 1: Create Lab5 package and type this class under Lab5 package:



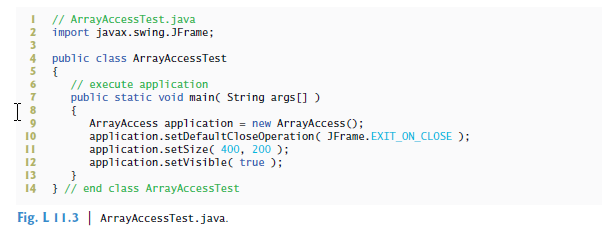
2. Also type this class under the same package:









1. Type this class to test the ArrayAcess class:
2. Following the comments in ArrayAccess class, insert the needed code to make it run as specified in the sample output above.

**Submit \*.java files to Lab5 folder by Sunday midnight at the latest.**