

CS 5000 – Summer 2025
Assignment #12, 50 Points
Bonus Assignment
Exception Handling – Chapters 12

Please remember to read and apply coding practices outlined in assignment 8.

Develop a complete Java program for each of the following problems. Please name the programs as indicated and add proper program headers and output labels as shown below. **Please use only concepts and programming constructs/syntax we discuss to date.** Make sure you **include a header for each program (see previous assignments).**

Program #1 (25 points): Design and implement a Java class, named *MismatchException*, to determine the average of four integer grades between 0 and 100. Please Do Not read user inputs as strings, user may enter one integer value per line. The program must *catch-and-try* exception handling block to detect invalid inputs (*InputMismatchException*). If an invalid input value is detected, the program prompts the user to re-enter four valid grades. The following sample test data shows input prompts and output labels and format. The sample test data below assumes that the user enters all numbers on one line as shown. User may enter one value per line. Make sure your code displays the outputs following the test data format.

First test: Enter four grades between 0 and 100: 10 20 30 40
Entered grades are: 10, 20, 30, 40
Student Average: 25.00

Would you like to run the program again (y/n)? y

Second test: Enter four grades between 0 and 100: 20 75 90 97.50
Incorrect input, re-enter valid grades: 20 75 90 g
Incorrect input, re-enter valid grades: 20 75 90 ;
Incorrect input, re-enter valid grades: 20 75 this 97
Incorrect input, re-enter valid grades: 20 75 90 97
Entered grades are: 20, 75, 90, 97
Student Average: 70.50

Would you like to run the program again (y/n)? y

Third test: Enter four grades between 0 and 100: 90 90 90 90.0
Incorrect input, re-enter valid grades: 90 90 90 90
Entered grades are: 90, 90, 90, 90
Student Average: 90.00

Would you like to run the program again (y/n)? y

Document your code, use proper prompts, format outputs as shown, use sound coding practices we learned thus far, do not hard code inputs, and allow program re-runs with different inputs (sentinel loop).

Program #2 (25 points): Design and implement a Java class, named *OutOfBoundException*, to determine and handle illegal array index value. The program defines an integer array of size 10 elements, populates the array with random integer numbers between 0 and 100, then asks the user to enter a valid index value between 0 and 9, and then it prints out the number stored in that

location in the array. The program must use *catch-and-try* exception handling block to detect out of bound index values (*IndexOutOfBoundsException*) and invalid input type (see program #1 above). If an invalid index or invalid input type is detected, the program prompts the user to re-enter a valid index value. This means 2 catch blocks, one for each exception handler. The following sample test data shows input prompts and output labels and format. Make sure your code displays the outputs following the test data format.

First test: Enter valid index between 0 and 9: 3
 The element at index 3 is 22

 Would you like to run the program again (y/n)? y

Second test: Enter valid index between 0 and 9: 23
 Index out of bound, re-enter a valid index: 56
 Index out of bound, re-enter a valid index: fr
 Index out of bound, re-enter a valid index: 1.4
 Index out of bound, re-enter a valid index: 8
 The element at index 8 is 87

 Would you like to run the program again (y/n)? y

Third test: Enter valid index between 0 and 9: 10
 Index out of bound, re-enter a valid index: 9
 The element at index 9 is 78

 Would you like to run the program again (y/n)? y

Document your code, use proper prompts, format outputs as shown, use sound coding practices we learned thus far, do not hard code inputs, and allow program re-runs (sentinel loop).

Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines posted in D2L. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment due date is posted in D2L.
3. Please compile, run, and test your code right before you upload your java files to the assignment submission folder in D2L.
4. Please upload only the .java files (total 2 files).