

PROGRAMMING ASSIGNMENT 5:

Read Savitch: Chapter 8 - Exceptions (and Chapter 6 - Arrays)
Programming: Name your file for PA5: **PA5.java** in the dir. named **PA5**

DUE: May 23, 2002 at 11pm

Turnin: **turninPA5**

REQUIRED:

This programming assignment will focus on basic Exception Handling within an application. You will perform I/O via keyboard and terminal with a `BufferedReader`. You will handle **`NumberFormatException`** and **`ArrayIndexOutOfBoundsException`** unchecked exceptions and declare (not handle) the checked exception **`IOException`**.

Program overview: The program will loop asking the user for the maximum number of scores the user wishes to enter. This must be an integer greater than zero. This value will be used to dynamically create an array of doubles. Then the user will enter some number of scores (one score per line) parsed as doubles and stored in the array. The user will indicate no more scores by typing <Control-D> at the keyboard (may be less than the maximum number of scores indicated earlier). The average of these scores will be displayed. The program will loop back to ask the user for the maximum number of scores to be entered again and repeat the input process and display the average. To quit the program, the user types <Control-D> when asked for the maximum number of scores to enter.

class PA5

```
private double[] scores;  
private int numOfScores;
```

```
public static void main( String[] args ) throws IOException  
public void calculateAverage( BufferedReader in ) throws IOException  
public void readScores( BufferedReader in, int arraySize ) throws IOException
```

main() will first open `System.in` as a `BufferedReader` (see Exceptions handout for an example of how to do this). Then create a `PA5` object to reference/access the instance members of this object. This is typical of applications starting in the static method `main()` needing to access their instance variables and methods. Then loop calling **`calculateAverage()`** passing the `BufferedReader`.

calculateAverage() will prompt the user to enter the maximum number of scores the user wishes to enter. This will be an integer. You will catch the **`NumberFormatException`** if the user enters an invalid integer value and notify the user with an error message (see the example PA5 for the format of the error messages) and loop back to ask the user to enter the maximum number of scores again. If the user types <Control-D> in Unix/Mac or <Control-Z> in DOS/Windows, an exit message is printed and the program exits.

If the user entered a valid integer, ensure that it is greater than zero. This value will be used in **`readScores()`** to dynamically create an array of doubles to store the entered scores. If the number entered is not greater than zero loop back to ask the user to enter the maximum number of scores again.

Once you have the a valid integer greater than zero, pass it to **`readScores()`** to create the array, read the scores from the user, and insert the valid scores into the array. After `readScores()` returns you are ready to calculate and display the average of the scores (see the example PA5 for the output format).

readScores() dynamically creates the array of scores (doubles) based on the `arraySize` parameter, prompts the user to enter scores, and loops reading scores (one score per line) inserting valid scores into the array until the user types <Control-D> when reading a score to indicate no more scores or the program tries to store a valid score into the array beyond the bounds of the array by catching the **`ArrayIndexOutOfBoundsException`**.

Catch **NumberFormatException** and **ArrayIndexOutOfBoundsException** and the catch-all **Exception**. In the case of **ArrayIndexOutOfBoundsException**, after displaying an error message return from `readScores()` to calculate the average of the (full) array. In the case of **NumberFormatException**, after displaying an error message loop back to allow the user to enter another score. In the case of the catch-all general **Exception**, display the exception parameter `[println(ex)]` and exit.

Examples: [Where you see ^D is where you hit <Control> and the D keys. It may not be echoed. Don't worry about that.]

java PA5

Maximum number of scores you wish to enter? **-9**
Must enter an integer > 0

Maximum number of scores you wish to enter? **0**
Must enter an integer > 0

Maximum number of scores you wish to enter? **8i8**
8i8 is not valid input.
Number of scores must be an integer. Try again.

Maximum number of scores you wish to enter? **5**

Enter Scores

^D

No scores were entered.

Maximum number of scores you wish to enter? **4**

Enter Scores

11.1

22.2

33.3

44.4

55.5

You entered more than 4 scores. No more allowed. Continuing ...

Average of the 4 scores entered is 27.75

Maximum number of scores you wish to enter? **5**

Enter Scores

11.1

22.2

33.3

^D

Average of the 3 scores entered is 22.20

Maximum number of scores you wish to enter? **4**

Enter Scores

5.5.5

5.5.5 is not a valid score. Try again.

44rf4

44rf4 is not a valid score. Try again.

55.5

66.6

99i4

99i4 is not a valid score. Try again.

^D

Average of the 2 scores entered is 61.05

Maximum number of scores you wish to enter? **^D**
Okay. Fine. Bye.