***Lab 9 –File Input/Output & Exception Handling***

***Answer the following questions.***

Instructor-led Demo:

Handle Exception:

1. Write a program that meets the following requirements:
   1. Create an array with one hundred randomly chosen integers.
   2. Cause an exception, *ArrayIndexOutOfBoundsException*, display the message “Out Of Bound”. You can display all the array elements using looping.

File Input/Output:

1. Given an array of integers, write a program that writes these integers into the file. Prompt the users to read the integers from the same file.

Exercise:

1. Write a program that counts the number of characters including words and lines in a file. The program prompts the user for inputting filename. Note that the IOException to be handled. Sample output as follows:

Please enter an integer value: 1

File Sample.txt has

1732 characters,

204 words and 70 lines.

*narrative.txt*

Java file input and out. Exception handling.Java file input and out. Exception

handling.Java file input and out. Exception handling.Java file input and out.

Exception handling.Java file input and out. Exception handling.Java file input

and out. Exception handling.Java file input and out. Exception handling.Java

file input and out. Exception handling.Java file input and out. Exception

handling.Java file input and out. Exception handling.Java file input and out.

Exception handling.Java file input and out. Exception handling.Java file input

and out. Exception handling.Java file input and out. Exception handling.Java

file input and out. Exception handling.Java file input and out. Exception

handling.Java file input and out. Exception handling.Java file input and out.

Exception handling.Java file input and out. Exception handling.Java file input

and out. Exception handling.Java file input and out. Exception handling.Java

file input and out. Exception handling.Java file input and out. Exception

handling.Java file input and out. Exception handling.Java file input and out.

Exception handling.Java file input and out.

1. Suppose that a text file **scores.txt** contains an unspecified number of scores. Write a program that reads the scores from the file and displays their total and average. Scores are separately by blanks.

**Hint**: Read the scores one line at a time until all the lines are read. For each line, use StringTokenizer or Scanner to extract the scores and convert them into double values using the Double.parseDouble method.

1. Write a program to create a file named **ints.txt** if it does not exist. Write one hundred integers created randomly into the file using text I/O. Integers are separated by spaces in the file. Using StringTokenizer or Scanner to read the data back from the file and display the sorted data.
2. Given the Loan class below:

|  |
| --- |
| 1. **package** loan; 2. **import** java.util.Date; 3. **public** **class** Loan { 4. **private** **double** annualInterestRate; 5. **private** **int** numberOfYears; 6. **private** **double** loanAmount; 7. **private** java.util.Date loanDate; 9. **public** Loan() { 10. // **TODO** Auto-generated constructor stub 11. } 12. **public** Loan(**double** annualInterestRate, **int** numberOfYears, 13. **double** loanAmount) { 14. **super**(); 15. **this**.annualInterestRate = annualInterestRate; 16. **this**.numberOfYears = numberOfYears; 17. **this**.loanAmount = loanAmount; 18. **this**.loanDate = **new** java.util.Date(); 19. } 20. **public** **double** getAnnualInterestRate() { 21. **return** annualInterestRate; 22. } 23. **public** **void** setAnnualInterestRate(**double** annualInterestRate) { 24. **this**.annualInterestRate = annualInterestRate; 25. } 26. **public** **int** getNumberOfYears() { 27. **return** numberOfYears; 28. } 29. **public** **void** setNumberOfYears(**int** numberOfYears) { 30. **this**.numberOfYears = numberOfYears; 31. } 32. **public** **double** getLoanAmount() { 33. **return** loanAmount; 34. } 35. **public** **void** setLoanAmount(**double** loanAmount) { 36. **this**.loanAmount = loanAmount; 37. } 38. **public** java.util.Date getLoanDate() { 39. **return** loanDate; 40. } 41. **public** **double** monthlyPayment(){ 42. **return** 0.0;//return actual monthly payment 43. } 45. **public** **double** totalPayment(){ 46. **return** 0.0;//return total payment 47. } 49. } |

Modify the Loan class to throw IllegalArgumentException if the loan amount, interest rate or number of years is less than or equal to zero.

1. Write a program to create your own exception class. In the main method, create an exception object of your exception class and handles the exception using try catch blocks.