Lab 9 –File Input/Output

Answer the following questions.

Instructor-led Demo:

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 the filename. Sample output as follows:

```
Please enter the filename: narrative.txt
File Sample.txt has
1732 characters,
204 words and 70 lines.
```

2. 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.

3. Write a program that removes a specified string from a text file. Your program reads the file and generates a new file without the specified string, copies the new file to the original file. Prompt the user for a string to be removed and the filename. For example, remove "Java" string in **datafile.txt**.

4. 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.

5. Write a program to create a file named **binaryint.dat** if it does not exist. If it exists, append new data to it. Write one hundred integers created randomly into the file using binary I/O.

6. Suppose a binary data file created in Q5 (**binaryint.dat**). Write a program to find the total of integers.

7. Write a program that stores an array of five `int` values 1, 2, 3, 4, and 5, a `Date` object for current time, and the double value 5.5 into the file named **objfile.dat.**

8. Given a *Loan.java* class. Rewrite the `Loan` class to implement `Serializable`. Write a program that creates five `Loan` objects and stores them in a file named **loanobj.dat.**

9. Given two files, write a program that concatenates these files and prints all content of these files. You are required to use SequenceInputStream class with its associated constructor as follows:

```
SequenceInputStream(InputStream s1, InputStream s2)
```

10. Give five files, modify the program in Q9 to read these files and print them on the console.

```
SequenceInputStream(Enumeration<? extends InputStream> e)
```