Assignment: Small Problems

Due: Sun. Nov. 21 at 5pm.

In this assignment, we are going to solve three small problems by using the knowledge of map, set/file, and Exception. Study the following problem descriptions and prepare your solutions for the three questions. Submit the solutions on the Blackboard website by Sunday, 21 November 2021.

For each program add sufficient java doc comments to explain the logic of the codes throughout the program. Use exception handler whenever appropriate. In addition, add screenshots that will show the output of each of the programs.

Question : Word Frequency [Frequency.java]

Write a program that reads the contents of a text file, named "words.txt". The program should create a map in which the keys are the individual words found in the file and the values are the number of times each word appears in the file. For example, if the word "the" appears 120 times, the map would contains an element with "the" as the key and 120 as the value.

The program should display the frequency of each word as output. Make sure to convert all uppercase words into lowercase before processing the count. In addition, punctuation characters should be removed from the sentences as well. The program should handle all file and map related exceptions. Iterators must be used to navigate the elements of the map variable.

Sample Run of the Program

Suppose the content of the file "words.txt" is the following:
Can you can a can as a canner can can a can?

Then output of the program would be the following:
can 6
you 1
a 3
as 1
canner 1

Work frequency rubrics | Total points: 10
[2] The program reads line of texts from a text file named words.txt
[2] The program cleans up the lines of text before calculating the count of words
[4] The program uses map variable to determine the count of each word
[2] The program uses exception handler throughout the code

Question : Dictionary [Dictionary.java]

In this program, first the program will read a text file named "dictionary.txt" that contains a number of valid text words with correct spelling. Next, the students would read words of texts from an input file named "words.txt". For each word read from the "words.txt" file, the program will check whether the word exist in the "dictionary.txt" file. If the word does not exist in the dictionary.txt file, then the program will display these words and the line/word number in the file so that they can be corrected.

Use a map variable to store all the words from the "dictionary.txt" file. Further, while reading each word from the "words.txt" file, verify whether the word exist in the map variable and based on that prepare the output of the program. Make sure to convert all uppercase words into lowercase before processing the count. In addition, punctuation characters should be removed from the sentences as well. The program should handle all file and map related exceptions. Iterators must be used to navigate the elements of the map variable.

Sample Run of the Program

First, suppose the content of the "dictionary.txt" file is the following:
name king once upon a time there was in the jungle of

Suppose the content of the file "words.txt" is the following:
Once upon a time, there was a bling in the jungle.
The names of the king was Drake Hunter.

Then output of the program would be the following:
bling 1 8
names 2 2
Drake 2 7
Hunter 2 8

Dictionary rubrics | Total points: 10
[2] The program creates a map variable by using the words of the file "dictionary.txt"
[2] The program cleans up the lines of text before calculating the membership of words
[4] The program uses map variable to determine the membership of each word
[2] The program uses exception handler throughout the code

Question : Set operations with Binary File Objects [SetOperation.java]

In this program, we are to create two text files by using the following class. Make changes in the following class as necessary.

```
class Student
{ private int st_id;
       private String name;
```

```
	public Student(int id, String name)
	{
		this.st_id = id;
		this.name = name;
	}

	int get_id() { return  st_id; }
	String get_name() { return name;}
}
```

The program will read values from a text file named "students.txt". In addition the program will open two files named "oddlines.txt" and "evenlines.txt" respectively.

The program will create objects by using the values from the read lines. Objects created by reading odd lines will be stored in the "oddlines.txt" file and objects that were created by reading the even lines will be stored in the "evenlines.txt" file. Note, the accessor functions can be used to store the object filed values in the respective text files.

Further, read the objects from the "evenlines.txt" file and store them in a set variable named, evenSet. Similarly, read the objects from the"oddlines.txt" file and store them in a set variable named, oddSet. The two set variables would be set of Student objects.

Lastly, display the objects that are common in both of these two sets by using set operations.

The program should handle all file and set related exceptions. Iterators must be used to navigate the elements of the set variable.

Sample Run of the Program
First, suppose the content of the "students.txt" file is the following:
5 John
5 John
2 Salma
2 Zayed
10 Sabiha
10 Sabiha
11 James
11 Jame

Then output of the program would be the following:
5 John
10 Sabiha

Set operation program rubrics | Total points: 10
[2] The program creates two files from the "students.txt" file [2] The program loads the read objects from the files in set variables [4] The program determines the common objects in eventSet and oddSet variables [2] The program uses exception handler throughout the code

Criterion	Details	Deductions
Classes	At minimum, we need the Frequency.java, Dictionary.java, and SetOperation.java files.	-10 x: The required Java files not submitted -5 x: The program does not provide the desired output -3 x: screenshot of the program run was not attached
Code quality	Identifier names, class names, proper use of public/private, ample comments in main, etc.	-15: incoherent, inconsistent coding style -10: comments were not used throughout the code - 5 x: exceptions were not handled throughout the code
Test	Note, whether the program compile and run	-30: The portion of the code of the assignment is a copy -10 x: The program does not compile