

Homework 3

(70 points)

Assigned: Thursday, February 5, 2026

Due: Sunday, February 15, 2026

Objective

This assignment is designed to practice hash-based and tree-based data structures, such as sets and maps.

Part 1. Set (20 pts)

Write a Java program (`Database.java`) that implements a simple database using the Java Set interface. The database stores unique names and supports basic operations such as adding, removing, searching, and printing names in ascending order.

Sample Program Input/Output

```
1. add
2. remove
3. search
4. print
5. Exit
1
Charles
1
Alice
1
Bob
1
Alice
2
Bob
3
Bob
not found
3
Charles
found
4
Total: 2
[Alice, Charles]
5
goodbye
```

Part 2. Map (50 pts)

Write a Java program (`WordFreqCount.java`) to read a text file (`book.txt`¹). Do not hardcode an absolute path in your program; assume the input file is in the same directory as your program. Count word frequencies using the Java `Map` interface. Display the top 20 most frequently occurring words as well as the least frequently occurring words. The order of the least frequently occurring words may vary depending on the implementation.

Detailed Instructions

When using a `TreeMap`, the keys are automatically sorted, but the values are not. You are required to implement a `sortByValue` method that sorts a map by its values and returns a list of map entries ordered by value. To sort a map by value, you must use the Java Collections sort method and implement your own `compare` method by defining a custom `Comparator`, rather than using the default comparator. For example, a general framework may look like the following:

```
public static List<Map.Entry> sortByValue(Map map) {
    .....
    Collections.sort(list, new Comparator<Map.Entry>() {
        public int compare(Map.Entry e1, Map.Entry e2) {
            .....
        }
    });
    .....
}
```

You are expected to properly handle any possible exceptions in your program.

Sample Program Input/Output

Top 20 Least Appeared Words:

```
(1): frowning --> 1
(2): instep --> 1
(3): abrupt --> 1
(4): Folk --> 1
(5): crisply --> 1
(6): Gang --> 1
(7): Eventually --> 1
(8): gloomy --> 1
(9): perspired --> 1
(10): spreading --> 1
(11): investment --> 1
(12): scratch --> 1
(13): require --> 1
(14): incites --> 1
(15): neatness --> 1
(16): Game --> 1
(17): teetotaler --> 1
(18): rounded --> 1
(19): role --> 1
(20): glade --> 1
```

¹Download from Blackboard.

Top 20 Most Appeared Words:

```
(1): the --> 5426
(2): I --> 3038
(3): and --> 2887
(4): to --> 2788
(5): of --> 2733
(6): a --> 2595
(7): in --> 1747
(8): that --> 1664
(9): was --> 1393
(10): it --> 1303
(11): you --> 1283
(12): he --> 1168
(13): is --> 1131
(14): his --> 1103
(15): have --> 908
(16): my --> 907
(17): with --> 849
(18): had --> 821
(19): as --> 780
(20): which --> 770
```

What to Submit

Submit your well-documented and properly formatted (indented) Java source files (`Database.java` and `WordFreqCount.java`) on Blackboard before the due date. You do not need to submit the input file (`book.txt`).