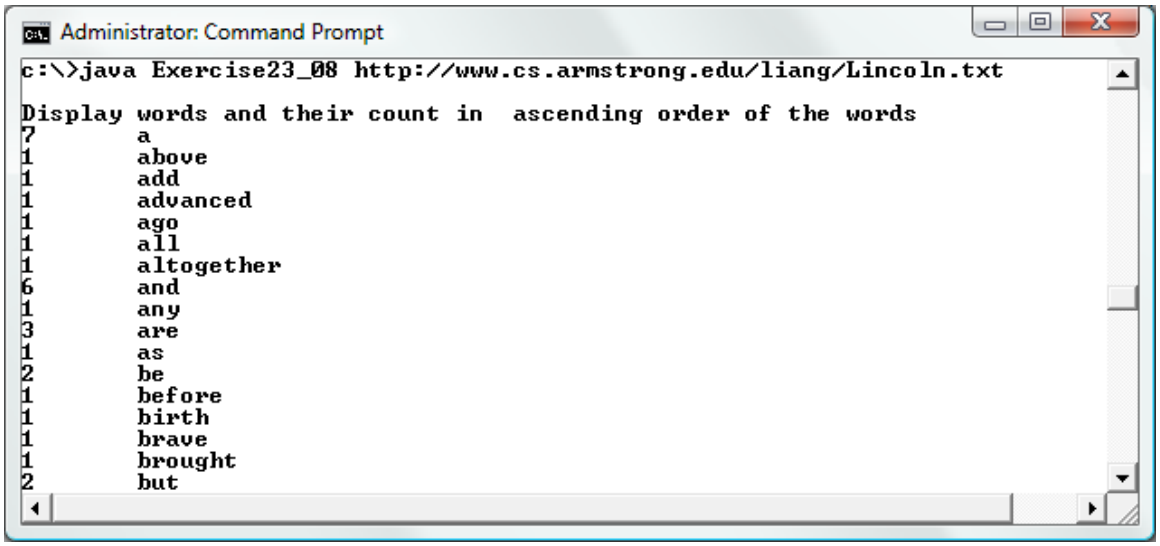
Student Name: _Amangeldi Madi_
Class and Section _CS-2104_____
Due: Monday, May 2, 2022 before 10AM

Data Structures and Algorithms
Astana IT University

Problem 1: Counting the Occurrences of Words

Problem Description:

Rewrite Listing 21.9 (from Daniel Liang's textbook) to read the text from a Web page. The URL for the Web page is passed as a command-line argument. Words are delimited by whitespace, punctuation marks (,;.:?), quotation marks ('"), and parentheses. Count words in case-insensitive fashion (e.g., consider Good and good to be the same word). The words must start with a letter. Display the output in alphabetical order of words with each word preceded by its occurrence count. Here is a sample run:



Please see page 510 for how to read data from a URL.

Analysis:
(Describe the problem including input and output in your own words.)

It's about reading an input from a text file on Web through URL. The code should count only words and print their occurrence numbers and the words themselves in alphabetical order.

Design:
(Describe the major steps for solving the problem. How do you use recursion to solve this problem.)

I did the code

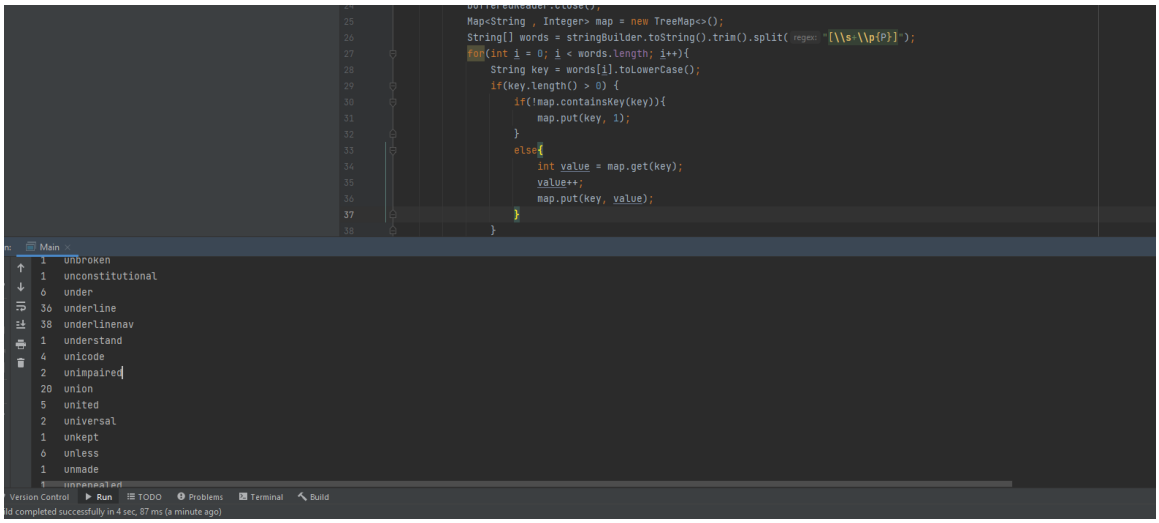
Coding: (Copy and Paste Source Code here. Format your code using `Courier 10pts`)

```
package com.company;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.*;
import java.net.*;

public class Main {
    public static void main(String[] args) {
        String text = args[0];
        System.out.println(text);
        try {
            URL url = new URL(text);
            BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(url.openStream()));
            StringBuilder stringBuilder = new StringBuilder();
            String inputLine;
            while((inputLine = bufferedReader.readLine()) != null){
                stringBuilder.append(inputLine);
                stringBuilder.append(System.lineSeparator());
            }
            bufferedReader.close();
            Map<String , Integer> map = new TreeMap<>();
            String[] words =
stringBuilder.toString().trim().split("[\\s+\\p{P}]");
            for(int i = 0; i < words.length; i++){
                String key = words[i].toLowerCase();
                if(key.length() > 0) {
                    if(!map.containsKey(key)){
                        map.put(key, 1);
                    }
                    else{
                        int value = map.get(key);
                        value++;
                        map.put(key, value);
                    }
                }
            }
            map.forEach((k, v) -> System.out.println(v + "\t" + k));
        }
        catch (java.net.MalformedURLException e) {
            System.out.println("Invalid URL");
        }
        catch (java.io.IOException ex) {
            System.out.println("IO Errors");
        }
    }
}
```

Testing: (Describe how you test this program)

Run the code

Screen Shot



Use https://liveexample-ppe.pearsoncmg.com/LiveRun/faces/LiveExample.xhtml?programName=ReadFileFromURL&username=slide&header=on to test your program and give me a screen using this test source.

```
12    try {
13      java.net.URL url = new java.net.URL(URLString);
14      int count = 0;
15      Scanner input = new Scanner(url.openStream());
16      while (input.hasNext()) {
17        String line = input.nextLine();
```

Enter input data for the program (Sample data provided below. You may modify it.)

```
https://liveexample.pearsoncmg.com/LiveRun/faces/data/Lincoln.txt
```

Compile/Run Reset Answer

Execution Result:

```
JDK8>javac ReadFileFromURL.java
Compiled successful

JDK8>java ReadFileFromURL
For security reasons, your program is not allowed to access files, directories, or URLs.

JDK8>
```

What to submit?

1. Submit PDF to Moodle (you must submit the program regardless whether it complete or incomplete, correct or incorrect)

3. Fill in self-evaluation:

1. Can your program display words only? Yes
2. Can your program display words in increasing order? Yes
3. Can your program count words correctly? Yes
4. Can your program pass argument from the command-line? Yes

Problem 2: Display nonduplicate names in ascending order

Problem Description:

> Given one or more text files (create yourself the txt file), each representing a day's attendance in a course and containing the names of the students who attended the course on that particular day, write a program that displays, in ascending order, the names of those students who have attended at least one day of the course. The text file(s) is/are passed as command-line argument(s).

Analysis:
(Describe the problem including input and output in your own words.)

Read the text as a command-line argument and display the names of each student with their attendance.

Design:
(Describe the major steps for solving the problem)

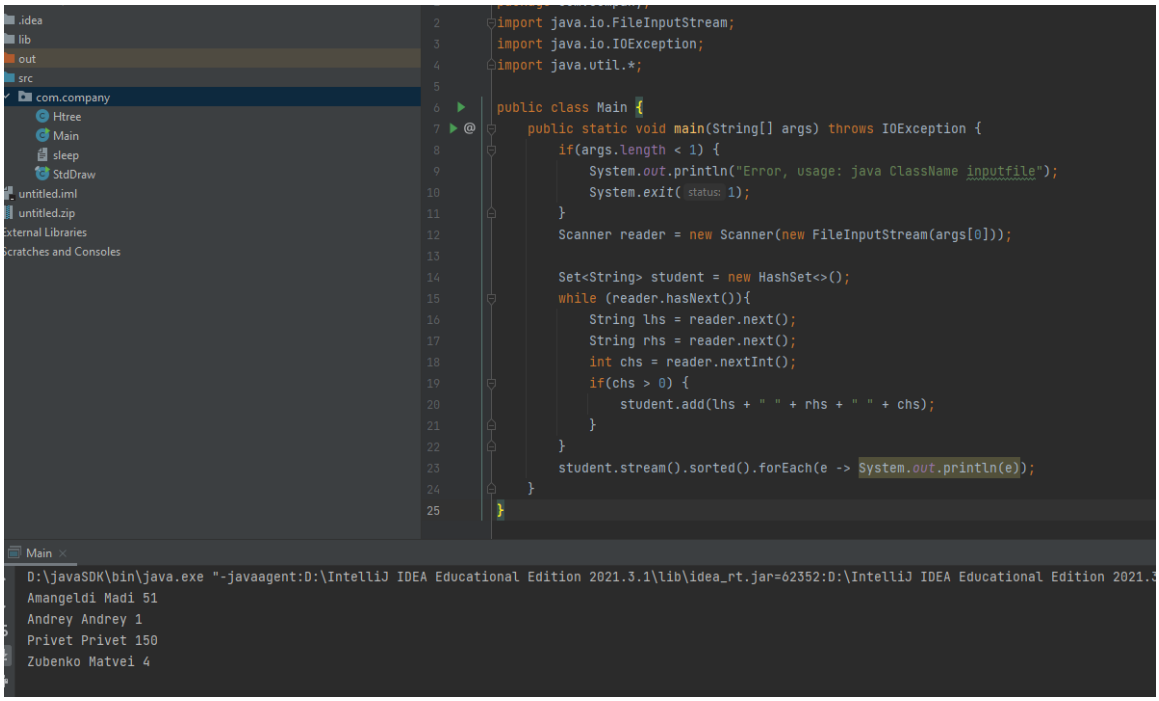
Creating hash set for storing students' attendance without duplicate names

Coding: (Copy and Paste Source Code here. Format your code using `Courier 10pts`)

```
package com.company;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.*;

public class Main {
    public static void main(String[] args) throws IOException {
        if(args.length < 1) {
            System.out.println("Error, usage: java ClassName
inputfile");
            System.exit(1);
        }
```

```
        Scanner reader = new Scanner(new FileInputStream(args[0]));

        Set<String> student = new HashSet<>();
        while (reader.hasNext()){
            String lhs = reader.next();
            String rhs = reader.next();
            int chs = reader.nextInt();
            if(chs > 0) {
                student.add(lhs + " " + rhs + " " + chs);
            }
        }
        student.stream().sorted().forEach(e -> System.out.println(e));
    }
}
```

Screen Shot:



What to submit?

1. Submit PDF to Moodle (you must submit the program regardless of whether it complete or incomplete, correct or incorrect)
2. Txt file you created

3. Fill in self-evaluation:

1. Can your program display words in ascending order the names of those students who have attended at least one day of the course? Yes

Problem 3: Checking whether a key exists in a set

Problem Description:

> Given a problem that is identical to the previous **Problem 2**, write a program that displays, in ascending order, the first names of those students who have attended more than one day of the course. If there are students with the same first name, display their first and last names.

Analysis:
(Describe the problem including input and output in your own words.)

Same but more than one instead of more than zero

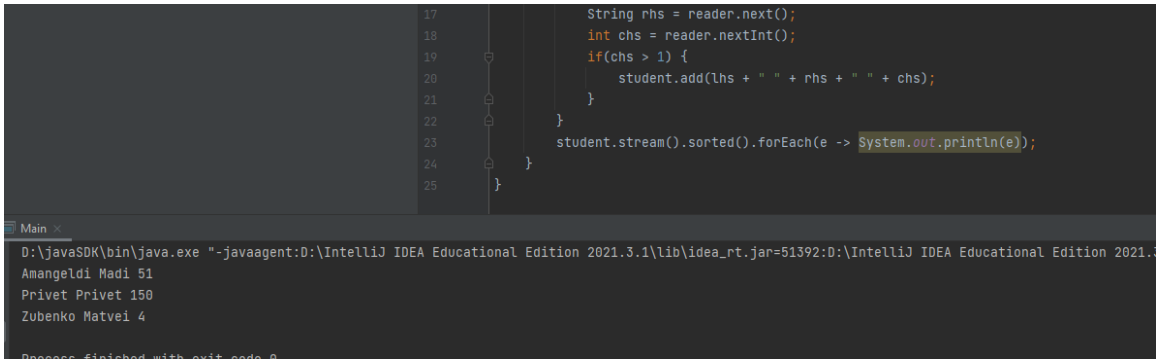
Design:
(Describe the major steps for solving the problem.)

Changing my if statement from >0 to >1

Coding: (Copy and Paste Source Code here. Format your code using `Courier 10pts`)

```
package com.company;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.*;

public class Main {
    public static void main(String[] args) throws IOException {
        if(args.length < 1) {
            System.out.println("Error, usage: java ClassName
inputfile");
            System.exit(1);
        }
        Scanner reader = new Scanner(new FileInputStream(args[0]));

        Set<String> student = new HashSet<>();
        while (reader.hasNext()){
            String lhs = reader.next();
            String rhs = reader.next();
            int chs = reader.nextInt();
            if(chs > 1) {
                student.add(lhs + " " + rhs + " " + chs);
            }
        }
        student.stream().sorted().forEach(e -> System.out.println(e));
    }
}
```

Screen Shot:



What to submit?

1. Submit PDF to Moodle (you must submit the program regardless of whether it complete or incomplete, correct or incorrect)
2. Txt file you created

3. Fill in self-evaluation:

2. Can your program displays, in ascending order, the first names of those students who have attended more than one day of the course? Yes