CSE 174 – Fall 2021

Assignment 2: 50 points – Due Thursday, Oct 7, 2021 by 11:59pm Ohio time

Outcome:

Students should demonstrate an ability to:

- Problem solving
- Write a Java program that can work with files
- Write a Java program that displays information to the console according to a given set of formatting guidelines
- Write Java source code that follows a given set of style guidelines

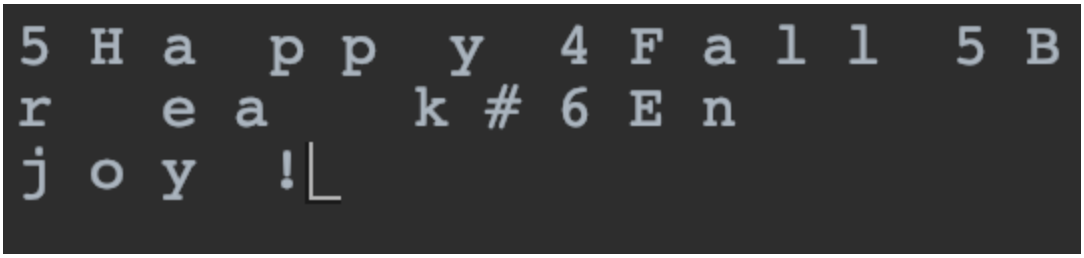
Application:

This application will ask the user to provide the name of an input file that contains numbers followed by characters. Then, it extracts the words and prints them on the display and inside a given file by the user.

Input File:

The Text File

- Download m1.txt file.
- Drag and drop the file on jgrasp to see what you have inside the file.



- Each number inside the file is followed by a bunch of characters. Those characters belong to a word and each number shows the number of characters for the next word. Example:
  The file starts with 5 showing the next 5 characters "H", "a", "p", "p", "y" belong to the first word which is "happy". The second number is 4 meaning the next 4 characters make the second word which is "Fall", and so on.
- You are to extract those words and print them all separated by a space on the display and in a file.
- As you are printing words, you need to print the next word on a new line whenever there is a "#" character.

Instructions Continue on Next Page!

Sample Runs #1:

```
Enter an input file name: m1.txt
Happy Fall Break
Enjoy!
**Do you want to print in a file (y/n): y
Enter an output file name: out1.txt
Printed inside the out1.txt successfully!
End!
```

```
Enter an input file name: m1.txt
Happy Fall Break
Enjoy!
**Do you want to print in a file (y/n): n
End!
```

Output file:

- As demonstrated on the sample runs your program should ask the user whether or not the user wants to print the result inside a file.
- If the answer is yes, then your program should:
    1. ask the user to enter a filename
    2. print everything inside the given file
    3. print a message showing it's done successfully
    4. end the program.
- If the answer is no, then your program just ends the program.

Instructions Continue on Next Page!

Input Checking:

- When your program asks the user whether the result should be printed in a file, the only two acceptable values are “y” and “n”. If the user enters anything else, then your program should ask the user again and again until the given answer is either “y” or “n”.

```
Enter an input file name: m1.txt
Happy Fall Break
Enjoy!
**Do you want to print in a file (y/n): yesss

**Do you want to print in a file (y/n): hi

**Do you want to print in a file (y/n): y
Enter an output file name: out1.txt
Printed inside the out1.txt successfully!
End!
```

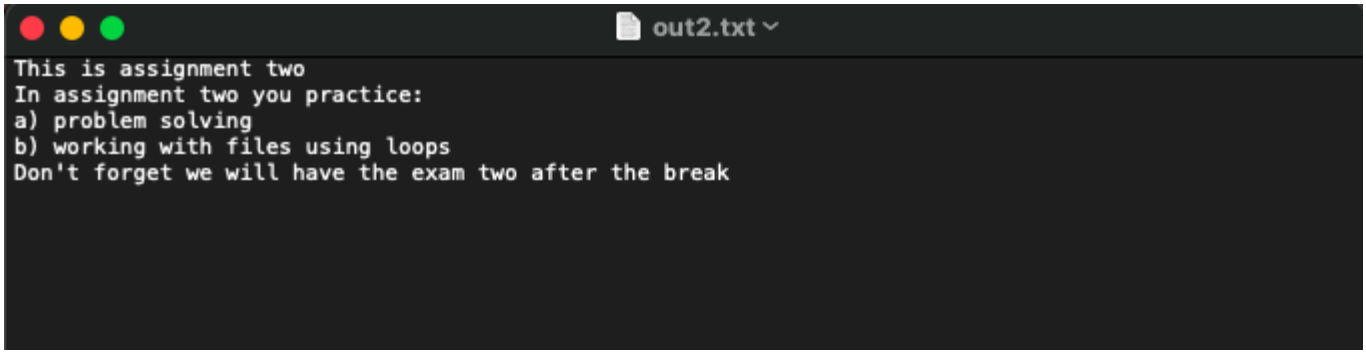
```
Enter an input file name: m1.txt
Happy Fall Break
Enjoy!
**Do you want to print in a file (y/n): no

**Do you want to print in a file (y/n): No

**Do you want to print in a file (y/n): n
End!
```

Instructions Continue on Next Page!

Testing & Sample Runs #2:

Download m2.txt to test your code with as shown in the sample runs, and make sure your code generates the same results.

```
Enter an input file name: m2.txt
This is assignment two
In assignment two you practice:
a) problem solving
b) working with files using loops
Don't forget we will have the exam two after the break
**Do you want to print in a file (y/n): y
Enter an output file name: out2.txt
Printed inside the out2.txt successfully!
End!
```




```
Enter an input file name: m2.txt
This is assignment two
In assignment two you practice:
a) problem solving
b) working with files using loops
Don't forget we will have the exam two after the break
**Do you want to print in a file (y/n): n
End!
```

Instructions Continue on Next Page!

```
Enter an input file name: m2.txt
This is assignment two
In assignment two you practice:
a) problem solving
b) working with files using loops
Don't forget we will have the exam two after the break
**Do you want to print in a file (y/n): N

**Do you want to print in a file (y/n): Y

**Do you want to print in a file (y/n): no

**Do you want to print in a file (y/n): Yes

**Do you want to print in a file (y/n): n
End!
```

Tips:

START EARLY!!! DO NOT WAIT UNTIL THE LAST MINUTE!!!! WE MADE THIS TIP AS ANNOYING AS POSSIBLE TO LOOK AT FOR A REASON!!!

- Always use **meaningful** variable **names**.
- Always start your code with **comments** at the top with at least your **name, your section and some description of the code**. Also add **comments inside** your **code** as well. The comments inside your code should explain what is happening on specific lines to help the reader understand your code.
- Finish your program step by step. Run your code each time you add something and make sure everything works as expected. **Do not wait until the last step to test your code!**

Instructions Continue on Next Page!

Your submission will be checked for potential academic dishonesty violations.

Submit your code on Canvas:

If your code is generating the same results as the sample runs, now you are ready to submit your code on canvas. If there are any style errors, fix them and resubmit your code.

Scoring:

- Submission by email is not acceptable. Whether that be code or screenshots, it will be a 0.

	Full Credit	No Credit or Partial Credit
Successful Submission via CODE (20 Points)	A fully successful submission to CODE that passes all of the required tests (2 tests) will earn full credit.	If your submission is not accepted by code, you will receive no or partial credit.
Additional Test Cases (10 Points)	There is 1 extra test case that is not required for your submission to be accepted; correctly handling it will earn full credit.	If you are not able to handle the extra test case, you will receive no credit.
Correct Style (10 Points)	You followed all of the given formatting requirements (indentation, **comments**, upper/lowercase, etc).	You did not follow some or all of the formatting requirements as specified in the requirement.
Proper Programming Practices (10 Points)	You used meaningful variable names. You used proper conditions and loops. You avoid repeating unnecessary parts of your code. You did not use things that were not discussed in the class.	Your code is structured poorly and you used things that were not discussed in the class.