Text Formatter User Guide

CSE 360 - Dr. Calliss - 85141

Fall 2019

Jose Elenes
Albert Haddad
Caden Midkiff
Madhu Madhavan Sridhar

TABLE OF CONTENTS:

Introduction:

The following guide provides insight into the use of a text formatting program that takes a text file as input and gives a formatted text file as output. The table of contents provides an index to relevant information to the user and can help with using the program. There is a list of commands and examples of their function. The goal of this guide is to clearly explain the functionality of the program and possible errors to users. The user guide includes a cover page. The cover page includes the title for the document, class semester and team members that participated in the design of the software. The second page has a table of contents. The table of contents has a list of pages and information associated with a topic in the user guide and this will be used to look for information necessary to understand what the software does. The table of contents includes section names and page numbers. The page is numbered after page two which is the table of contents.

The Introduction starts on page three. It describes the document and helps to navigate its table of contents correctly. The Introduction is followed by external requirements. The external requirements discussed what is required to execute the program. This is followed by a getting started section which includes a screenshot of the GUI at the start of the program and a brief description on how to get started formatting text. After this section there is a description of all formatting options and examples of how these options affect the text. This is followed by a description of possible error messages and instructions to fix the error. The guide ends with an explanation of how to end the program.

Overview:

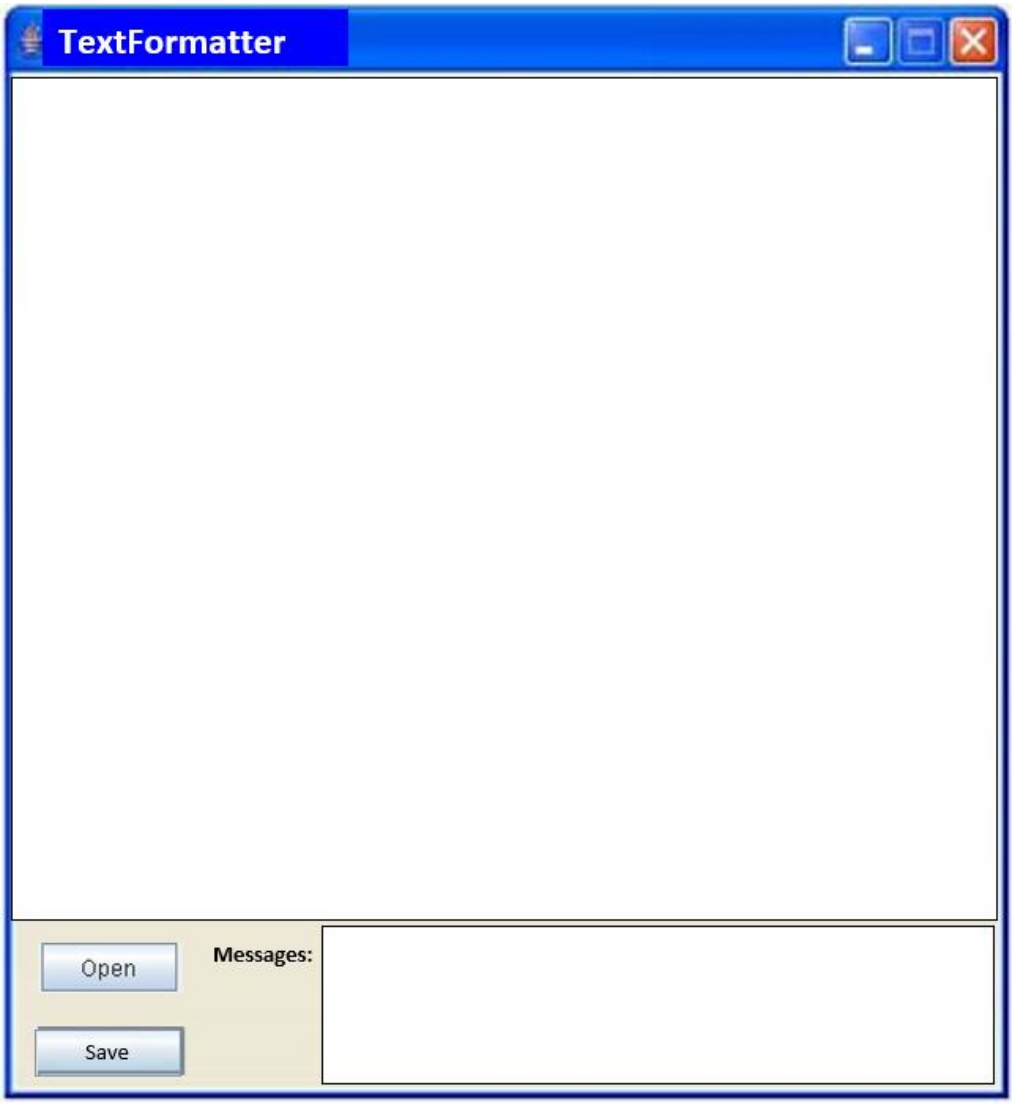
The Text Formatter Tool will take an input file, process the file, then output a text file formatted according to the commands in the input file. The input file must be a .txt file and must contain only valid characters and commands. The output file removes all commands, leaving only the formatted text. This text can be previewed inside the GUI and will also be saved to the PC as a text file. The default formatting will be a line length of 80 characters, left justified, no wrapping, no indentation for each new paragraph, single spaced and will be printed in one column. All of these default format options can be changed. A full list of commands can be found in the command section. Words will never be cut off halfway through. Commands must each be written on a separate line, and commands will only affect the text that is found under that command. For example, a command on line 4 to right-justify a document would only right-justify lines starting with line 5.

The GUI for the text formatter tool can open and save a file with words. It also can handle errors and will display a short error message describing the problem. The text formatter tool was created using GitHub for version control and was written on Eclipse using JAVA code. A GUI template was used to generate the user interface, but the code driving the formatting of text was written from scratch with equal contributions from the four team members.

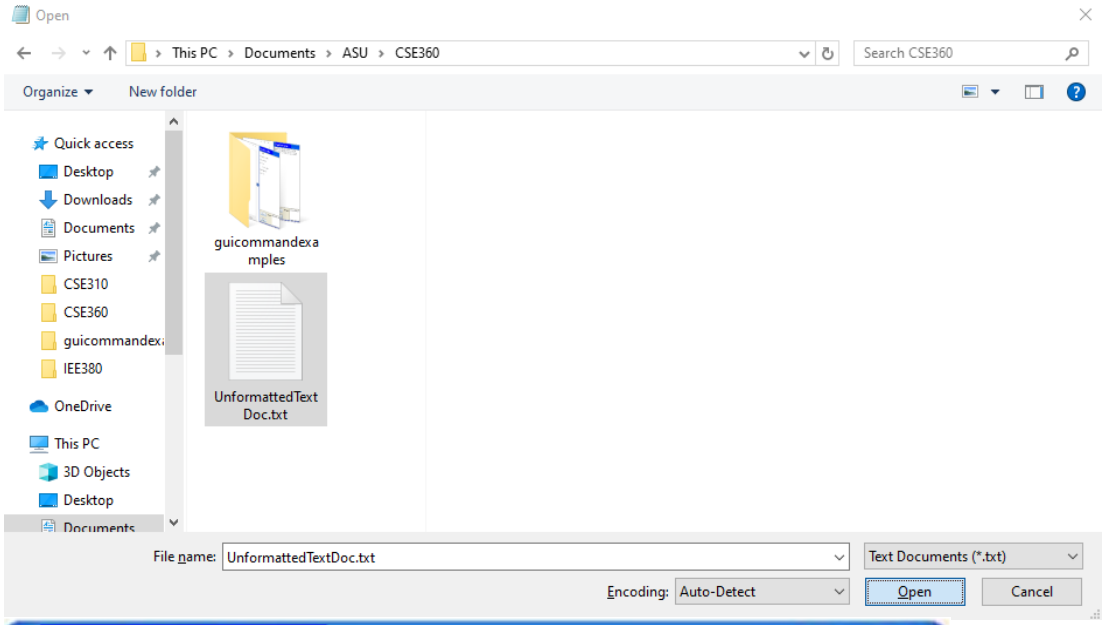
External Requirements:

The requirements to run this program include a computer system with java application running on it. Since the language is JAVA it can be deployed in any system. The program can also be integrated into a larger database and use its structures. The compiler can translate JAVA code into an executable by various systems. JAVA is a versatile language that can be a multi platform and therefore this program can work on MAC, LINUX, Windows operating systems. The program takes in a string of a minimum character size of 30 and therefore this can only work with strings. If any numbers are entered these are translated into strings and the same rule applies. The file is a text formatting program and it is restricted on the operation it can accomplish by string and string manipulations. A minimum configuration of a computer with Java application running on it and an input text file that requires formatting is the requirements to execute the program. The program will only run if all of its associated files accompanied are placed in the same folder and are accessible to the compiler. The permission on the file needs to be public and or modify the file permissions to transfer between different computers.

Starting view:



Upon opening the program, this window is what the user will first encounter. Both of the text boxes will be blank as the User has not tried to format any text yet. The user should have their unformatted text file ready to go at this point.



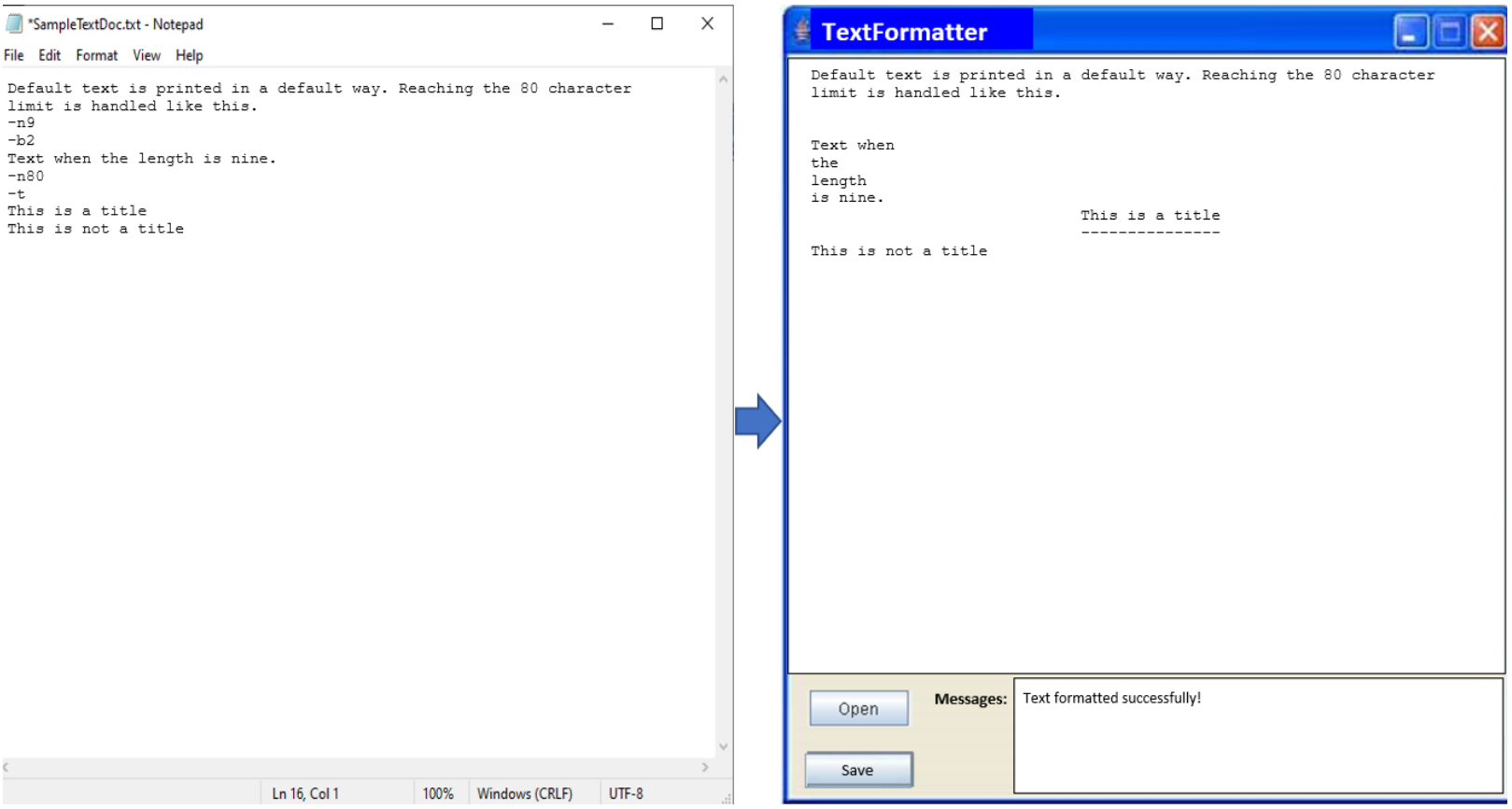
To begin, press the “Open” button. This will open a file explorer where the user can navigate their computer to find their text document. Once they find their document, they should click on the file, and click the “Open” button in the File Explorer. Once the program checks to ensure the proper type of file is selected, it will proceed to format the text. No other work from the user is needed until the program is finished formatting

Once the program has finished running, the user will see their formatted text in the “Text Display Area”, as highlighted on the left. This shows them a preview of what their text looks like, formatted. A message stating the program has finished formatting the text will appear in the “Messages Box”, as well as any errors the program might have ran into. From here, they have the choice to go back into their text file and change something that might not have turned out as they wanted it, and can run the file through the formatter following the same steps as before. However, if they are satisfied with how their file turned out, they can press the “Save” button to save their newly formatted file over their previous one. Upon doing this, a message stating that the file was successfully saved will be displayed in the “Messages Box”, and the user can close out of the program.

List of Text Formatting Commands:

-n#	Number of characters per line
-r	Right Justified Text
-l	Left Justified Text
-c	Center Justified Text
-e	Equal Spacing
-w+	Word Wrap On
-w-	Word Wrap Off
-s	Single spaced
-d	Double spaced
-t	Center title
-p#	How Much To Indent Each Paragraph
-b#	Number of blank lines to insert
-a#	Number of columns(restricted to 1 or 2)

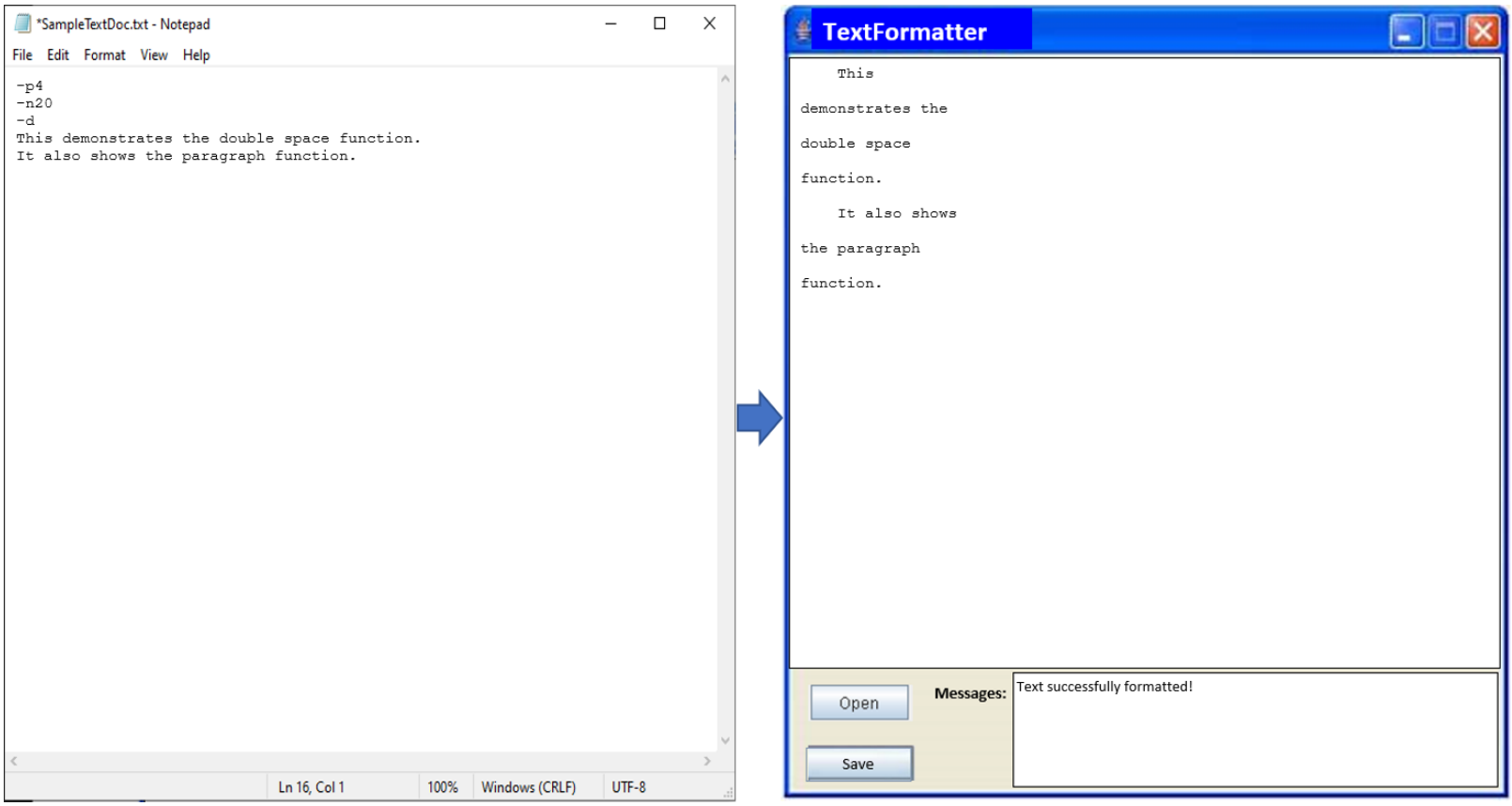
Examples of Commands In Use:



Characters Per Line (-n#): By default, the formatter will make each line a maximum of 80 characters before moving to a new line. By inputting the command "-n#", where # = the number of characters per line, the formatter will change how many characters there are per line to # only from that point forward until it is otherwise changed.

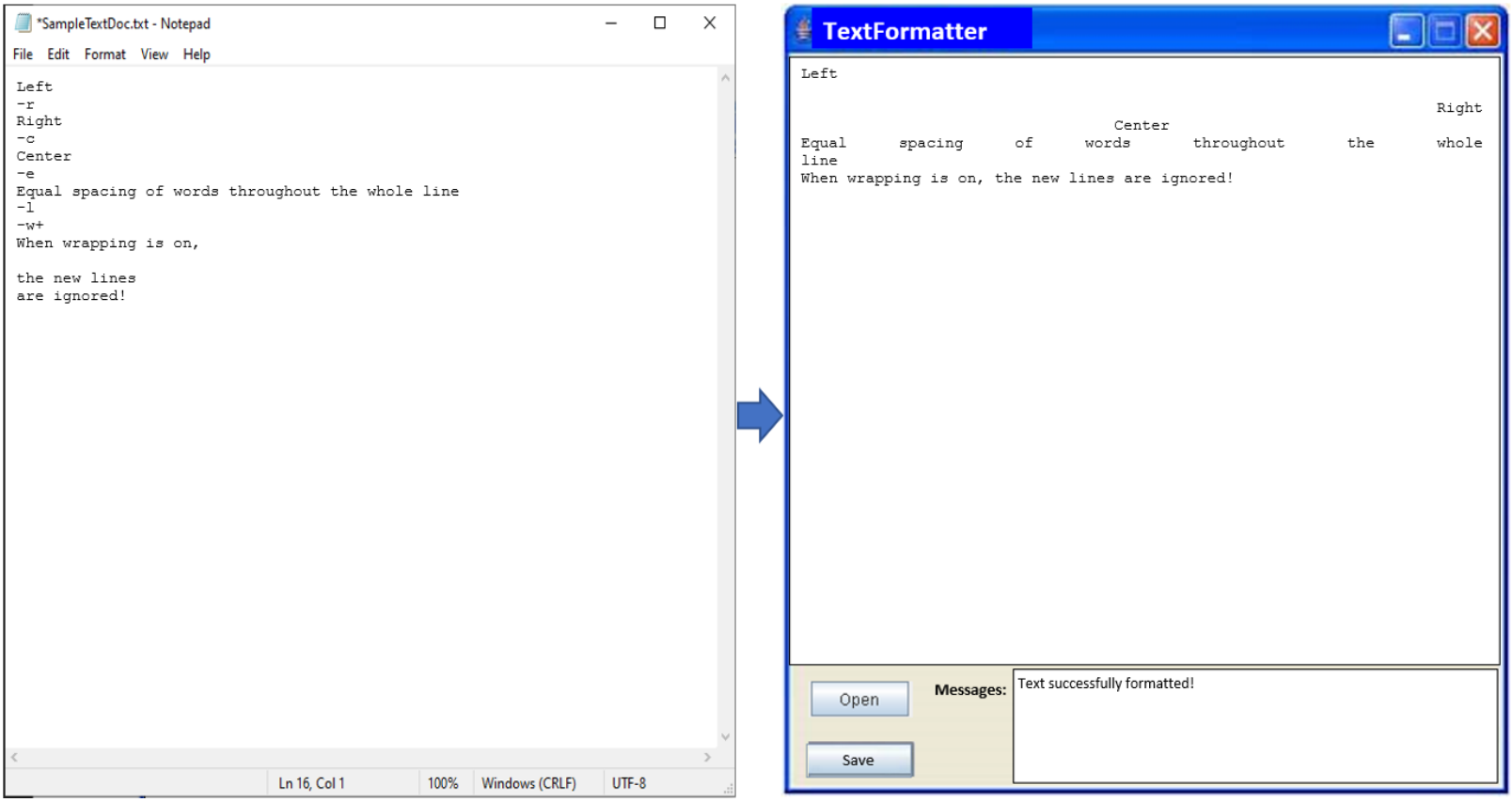
Blank Lines (-b#): If the user wants to add blank lines at any point in their file, they can write in the command "-b#" where # = the number of blank lines to add to the line under the command.

Titles (-t): Turns the line under the command into a title, which center aligns the line under the command, then creates a line of dashes under that line which is also center aligned and is the same length as the title line.



New Paragraph (-p#): When the user wants to start a new paragraph, they can input the "-p#" command, where # = the amount of spaces the user wants in front of the first word of the new paragraph, as to show indentation of a new paragraph.

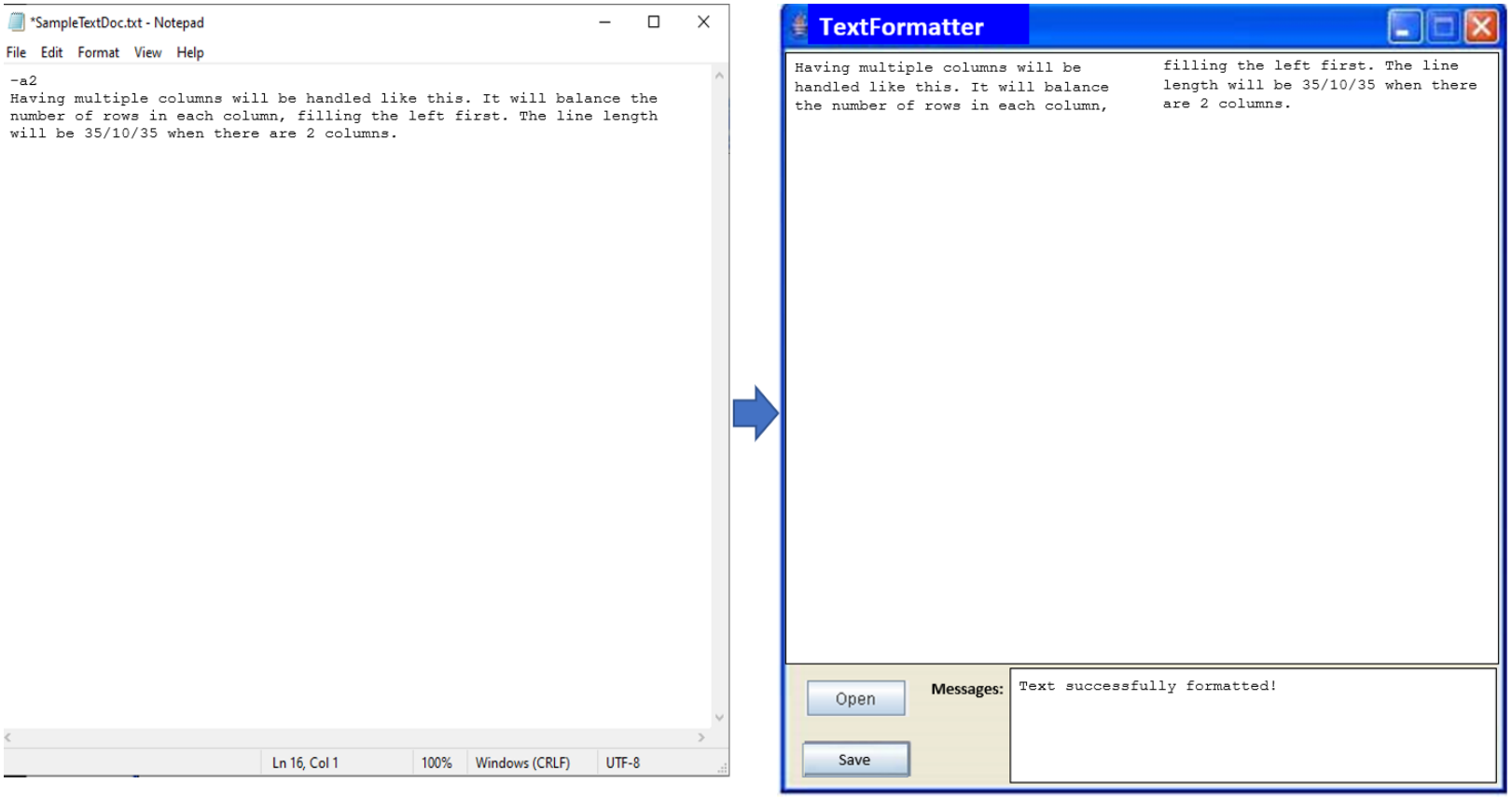
Line Spacing (-s, -d): By default, the formatter will assume that the user wants the line spacing to be single spaced. By inputting "-s" for single space, or "-d" for double space, the formatter will switch to that line spacing from the point of the command forward until otherwise specified.



Text Alignment (-l, -c, -r): These commands will align the text to either the left, center, or right side of the page, from the line that the commands are stated, for the rest of the document or until it is changed by commands .

Equal Spacing (-e): This command will change the lines starting from where it is stated until it is changed otherwise, to have equal spacing between the words so the lines span from the beginning of the line until the end.

Text Wrapping (-w+, -w-): This command will toggle the text wrapping feature of the program, the default of wrapping off, “-w-” which leaves all white space in the input text document as entered. Turning wrapping on will condense text if there is extra whitespace in the input file. New lines and extra spaces will be ignored so as much can be fit on one line as possible.

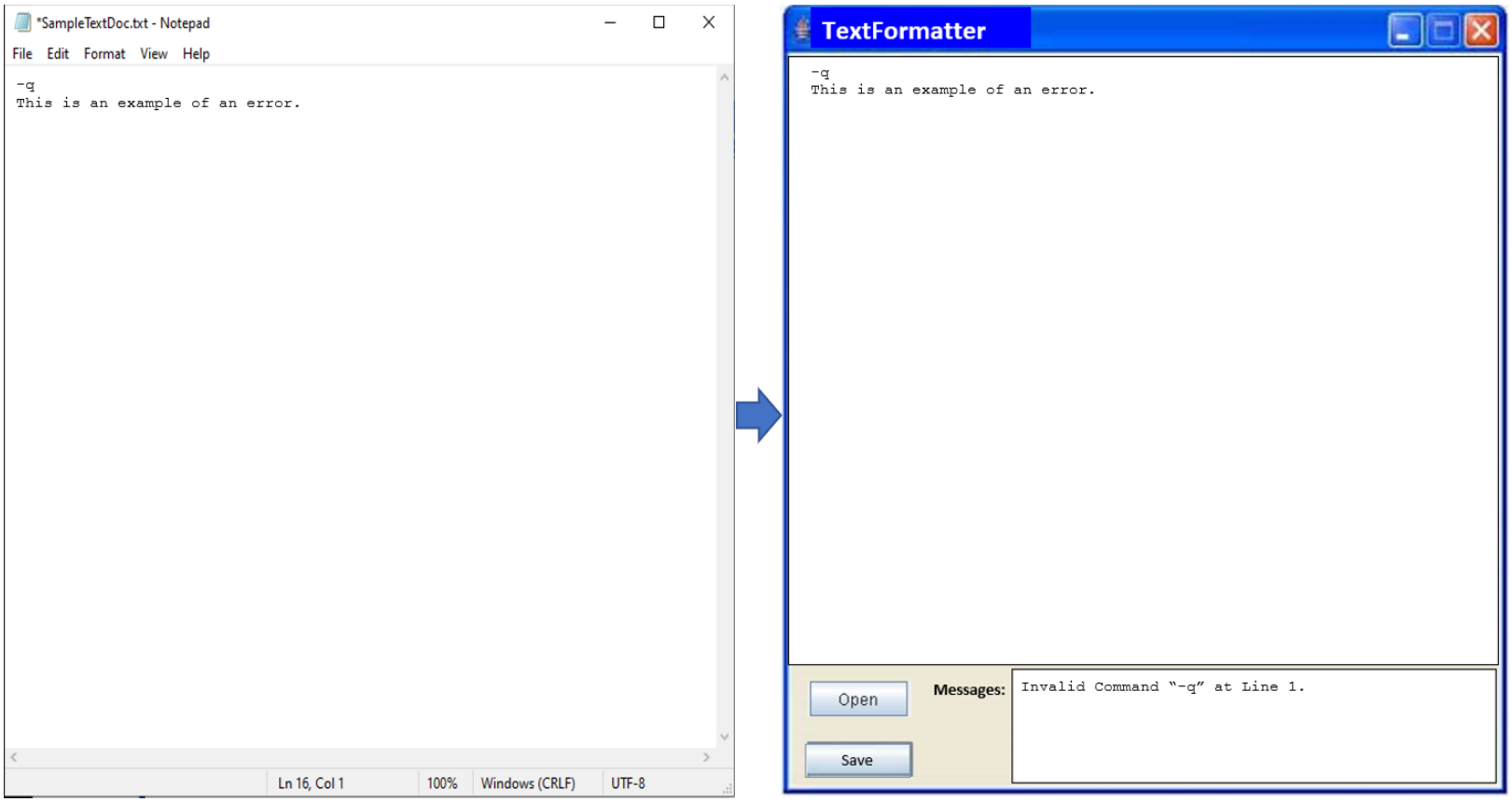


Columns (-a#): This command will turn the document into a 2 column document. The line length gets changed to 80, and the spacing on each line gets turned to 35 character width for the first column before the line gets wrapped down to the next line, a column of 10 empty characters to separate the columns, then another 35 characters that make up the second column.

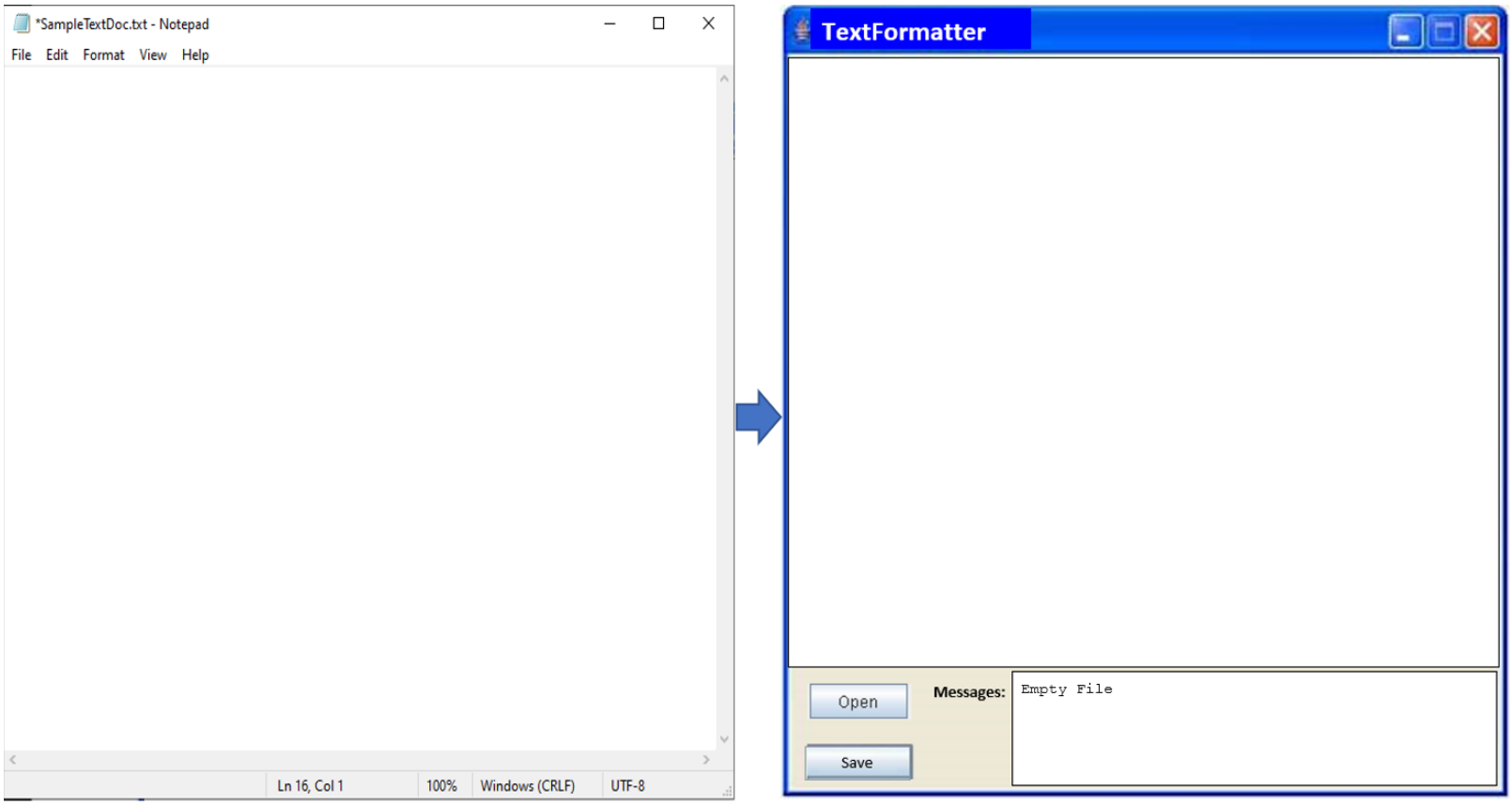
Examples of error messages & conditions:

If the program runs into any issues or errors, it will communicate what is wrong to the user in the Messages Text Box at the bottom of the window. Errors can stem from the user uploading an empty file, to a user trying to input an invalid command in their text file for the formatter to try to follow.

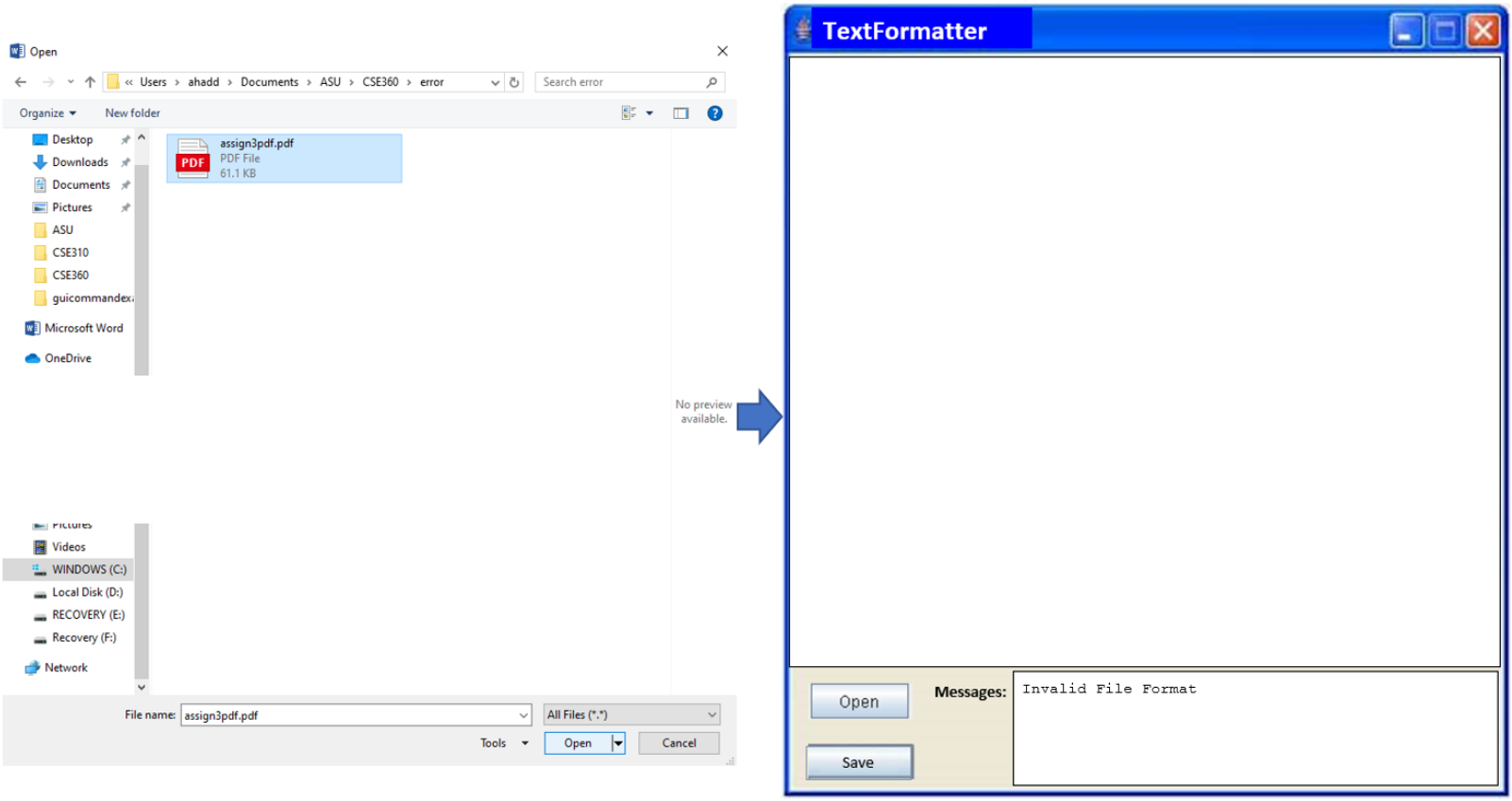
Error	Error Message	Explanation
Invalid file format	Please upload a .txt file.	The Input file must be in .txt format.
Empty File	The file is empty.	The Input file should not be empty.
Invalid Command	Please enter a valid command.	Only the defined commands should be used.



Invalid Command: The file was uploaded using "-q" as a formatting command. The command is not one that the program supports or recognizes, so it informs the user of the error, which command is at fault, and at which line in the document. This error can also occur if an invalid parameter is entered into a valid command, such as "-a3".



Blank File: This error will occur if the user tries to format an empty text document. There is nothing to format, so the program will notify the user that nothing was formatted because an empty document was uploaded.



Invalid File Format: This error will occur if the user tries to upload a file that is not a .txt file. The formatter is only able to work with a .txt file, so uploading any other file type will cause an error.

Ending the program:

Once the user has finished using the formatter, and would like to close the program, they can choose to press the “Save” button to save their newly formatted text if they are satisfied with the results, or is able to upload another text file for formatting. Once the user is finished, they can simply press the “X” button on the top right of the window to close the program.