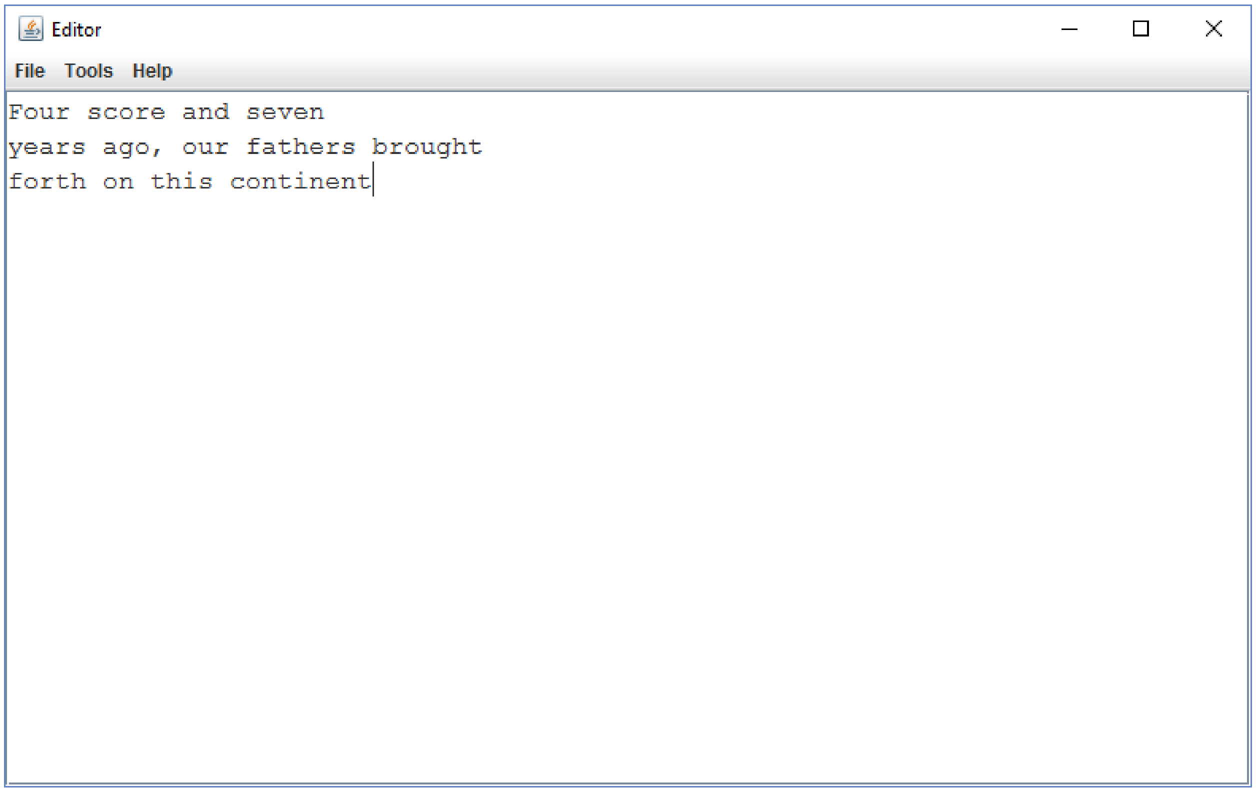
GUI & IO Application

ISTE-121 – Homework 02

Overview
This assignment creates a GUI application that allows a user to choose a file, load it, modify it, and save it to the original file or a new file.

Write code to display the following Editor interface. It is just a VBox with a MenuBar and a TextArea.



```
Four score and seven
years ago, our fathers brought
forth on this continent
```

The File menu has menu items for New, Open, Save, Save As, and Exit. The Tools & Help menus have just one menu choice each —“Word Count” & “About Editor” respectively.

Overall Design
Use two class files to create this application:

1. “FileListener” is an **external class** that contains an actionPerformed method that handles all the File menu choices
2. “Editor” contains the GUI constructor definition and a main that instantiates the class. The event listeners for the Tools and Help menu are in the Editor class as **anonymous inner classes**.

Detailed Design
The file menu handler will need to track the current file either with a file name string variable or with a File object variable.

<u>File menu actions:</u>

“New”:	Clears text area and name (or object) of current file. Sets the title bar to simply “Editor”.
“Open”:	Uses a FileChooser to obtain a file name from the user. It then loads the contents of the specified file into the TextArea and stores the file name (or object). If a file is successfully loaded, the title bar becomes “Editor: <NAME>” where NAME is the name of the file loaded.
“Save”:	Writes the contents of the text area to the file named in the title bar (the last file opened). If no file has not been named (e.g., if the file started with the New menu item), a message appears suggesting to use Save As. If a file is successfully saved, the title bar becomes “Editor: <NAME>” where NAME is the name of the file saved.
“Save As”:	Uses a FileChooser to obtain a file name from the user (use showSaveDialog instead of showOpenDialog). It then writes the contents of the text area to the specified file and stores the file name or object. If a file is successfully saved, the title bar becomes “Editor: <NAME>” where NAME is the name of the file saved.
“Exit”:	Ends the application without warning.

<u>Tools menu actions:</u>

“Word Count”:	Counts number of words in the text area. Consider only white space characters (`[\t\n ]`) as delimiters. Use an Alert to display the result as shown in the example below. The easy way to count words is to split the text in the Editor text area based on white space and use the number of strings returned as the word count.

<u>Help menu actions:</u>

“About Editor”:	Uses an Alert to display the author’s name and the date of completion, as shown below.

NOTE: in my solution, the TextArea is set to use a font called “Courier”, with a size of 16. Yours should do the same. To do so, use the Font.font("Courier", 16) method.

ALSO, the TextArea should be set to wrap text on a word boundary … see the setWrapText method in TextArea.

My Solution
You can download and run my solution to this homework from myCourses (found in HW2Downloads). It is named HW2.jar. Use this to compare my program to your solution. You will also find the file Declaration.txt in HW2Downloads. This is the declaration of independence, and will allow you to test word wrapping.

Submission:
Zip your java & class files and submit the zip file to the homework dropbox.

The homework grading criteria are shown on the next page. No points will be awarded if the program does not compile and run.

ISTE-121 Homework 2 gradesheet Name: ____________________________

Requirements	**Point Value**	**Points Earned**
Editor Class: • Creates GUI as shown • Editor main TextArea uses Courier font and size 16. • Registers appropriate handlers • Uses private methods to avoid repeating code or making handle too large (this should have placed under FileListener Class)	10 6 5 7	
FileListener Class: • Includes private attributes for handling the file, the text area, and the stage. • Handles "Exit" menu option properly and uses System.exit() • Handles "New" menu option properly and resets appropriate information • Handles "Open" menu option properly, uses Alert, reads from a file anywhere on user's system, and tracks the specified file • Handles "Save As" menu option properly, uses FileChooser, writes to a file anywhere on user's system, and tracks the specified file • Handles "Save" menu option properly writing to the previously specified file • Handles "Word Count" menu option displaying the result using an Alert • Handles "About Editor" menu option properly	3 2 10 10 10 10 10 3	
Anonymous inner classes for Word Count: • Uses Alert to display information • Displays correct information and uses correct format	2 2	
Anonymous inner classes for About: • Uses Alert to display information • Displays correct information and uses correct format	2 2	
General: • Uses Mnemonics for menu items • Uses menu separators as appropriate • Uses ellipses as appropriate	2 2 2	
Total:	100	
Deductions	**Point Value**	**Points Deducted**
- JavaDoc Header Comments - Section & variable Comments - White Space - Indentation - Variable Names - Method Names	0-5 0-5 0-5 0-5 0-5 0-5	
Total Points	**100**	

Comments: