Criteria for level 1

Specification

- The Program is a to-do list that allows the user to store tasks
- All of the tasks are stored on a separate file.
- The program receives input from the user, through keyboard,
- The data format in this code is text. It is stored in a text file named "todolist.txt" and read into an ArrayList of strings. The tasks are written and edited in the text file as separate lines of text.

Correctness and exception handling

- There are 5 cases possible; the expected responses are as follows;

Case 1: Add a new task
- If no text is already stored, the text is put in the first line if otherwise on a new line.

Case 2: Clear a specific task
- If option 2 is chosen it will then ask prompt you to input the line number which you want to clear, if the line is to be cleared a simple press of the enter button will do so.

Case 3: Marking a task as done
- If prompted, the code will add a " - done " marking to the end of the line

Case 4 : Clearing the whole list, if chosen all the text from the file will be cleared
Case 5 : Exit, the program will terminate with exit code 0

An additional .txt file is provided

Additional criteria for level 2

Correctness and exception handling

- FileNotFoundException - If the file is not found, the code prints "File not found.".

Additional Criteria for Level 3:

Correctness and exception handling or wow-factor

These cases are not handled, as they are unlikely to pop up and would not cause significant errors in the usage of this code:

1. NoSuchElementException - If there is no next element in the scanner, this exception is thrown.
2. IllegalStateException - If the scanner is closed and the next method is called, this exception is thrown.

Resource management

- The file resource is released when it is no longer needed by using a try-with-resources statement. This ensures that the file is closed even if an exception is thrown during its use.