File Handling in Java: Reading, Writing, and Appending

Introduction:

File handling is a crucial aspect of programming that involves manipulating files stored on a computer's filesystem. In Java, file handling operations include reading from, writing to, and appending data to files. These operations are performed using classes provided by the **java.io** package.

Objective:

The objective of this Java program is to demonstrate file handling operations - reading, writing, and appending - through a simple console-based application.

Implementation:

1. **User Input:** The program starts by prompting the user to enter the name of the file (including its extension) they want to work with.
2. **Writing to File:** After receiving the file name, the program asks the user to input the initial content they want to write to the file. The **writeToFile** method then writes this content to the specified file.
3. **Reading from File:** Next, the program reads the content of the file using the **readFromFile** method and displays it to the user.
4. **Appending to File:** The program prompts the user to input additional content they want to append to the file. The **appendToFile** method appends this content to the existing file.
5. **Displaying Updated Content:** Finally, the program reads the content of the file again after appending and displays the updated content to the user.

Conclusion:

This Java program provides a basic understanding of file handling operations, allowing users to interactively read, write, and append data to files. It serves as a foundation for more complex file manipulation tasks in Java applications.

Algorithm: -

1. Start the program.
2. Prompt the user to enter the name of the file (including its extension).
3. Read the input file name from the user and store it.
4. Prompt the user to enter the initial content they want to write to the file.
5. Read the initial content from the user and store it.
6. Write the initial content to the file specified by the user.
7. Read the content of the file and display it to the user.
8. Prompt the user to enter additional content they want to append to the file.
9. Read the additional content from the user and store it.
10. Append the additional content to the file.
11. Read the content of the file again after appending.
12. Display the updated content of the file to the user.
13. End the program.

This algorithm outlines the step-by-step process followed by the Java program for handling file operations, including reading, writing, and appending data to files.