🔹 4. File Stream Declaration

**ifstream inputFile("filename");**

**ofstream outputFile("filename");**

**Purpose**:

ifstream for reading from a file.

ofstream for writing to a file.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 5. Check if File Opened Successfully

**if (!fileStream.is\_open()) {**

**// handle error**

**}**

**Purpose**: Ensures the file was successfully opened before proceeding.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 6. Reading File Line by Line

**while (getline(input\_stream, string\_variable)) {**

**// use the line**

**}**

Full General Syntax:

getline(input\_stream, string\_variable, delimiter\_character);

input\_stream: usually cin or ifstream

string\_variable: the variable where the line is stored

delimiter\_character (optional): character that ends the read (default is newline \n)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 7. String Search

**string\_variable.find("some\_text") != string::npos**

**Purpose:** Checks if a substring exists in a string.

Returns position (index) of the first match, or string::npos if not found.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 8. String Replace

**string\_variable.replace(start\_index, length\_to\_replace, "new\_text");**

**Purpose:** Replaces part of a string.

start\_index: where replacement starts.

length\_to\_replace: number of characters to replace.

"new\_text": the new text to insert.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 9. Get Last Character of a String

**string\_variable.back()**

**Purpose:** Returns the last character of the string.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 10. Substring (Remove Last Character)

**string\_variable = string\_variable.substr(0, string\_variable.size() - 1);**

**Purpose:** Removes last character (typically a semicolon in this code).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 11. Writing to a File

**outputFile << "some text" << endl;**

**Purpose:** Writes data into a file stream with a newline at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 12. Closing Files

**inputFile.close();**

**outputFile.close();**

**Purpose:** Closes the files after reading/writing to free up system resources.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

🔹 13. Printing to Console

**cout << "text to print" << endl;**

**Purpose:** Outputs to terminal (standard output).