
◆ 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).