

**Name : Aavani Rajesh Perumbessi**

**Roll no : 15096724059**

**Cohort : Mark Zuckerberg**

**File Handling in C++**

# Introduction to File Handling



## What are files?

Files are fundamental building blocks for storing and accessing data in computer systems.



## C++ file handling

C++ provides robust mechanisms to interact with files, enabling you to read, write, and manipulate data stored on your device.

# Opening and Closing Files

## Opening files

Use ofstream for writing, ifstream for reading, and fstream for both.

## Closing files

Always close files using close() to release resources and ensure data integrity.

# Reading and Writing to Files

## 01 Reading

Use `get()`, `getline()`, and `read()` for various reading operations.

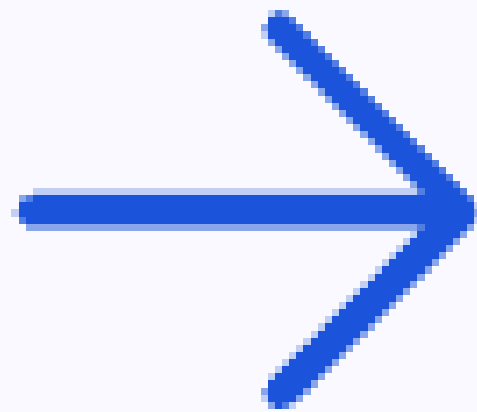
---

## 02 Writing

Employ `put()`, `write()`, and `operator<<` to insert data into files.

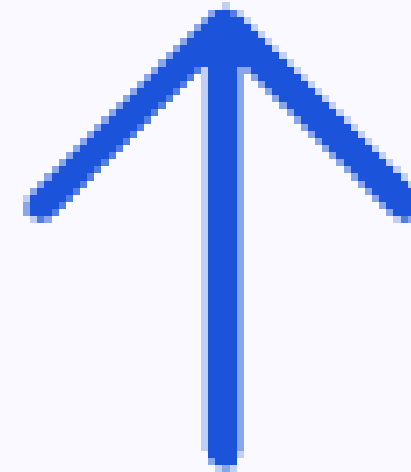
# Navigating Within Files

Seeking



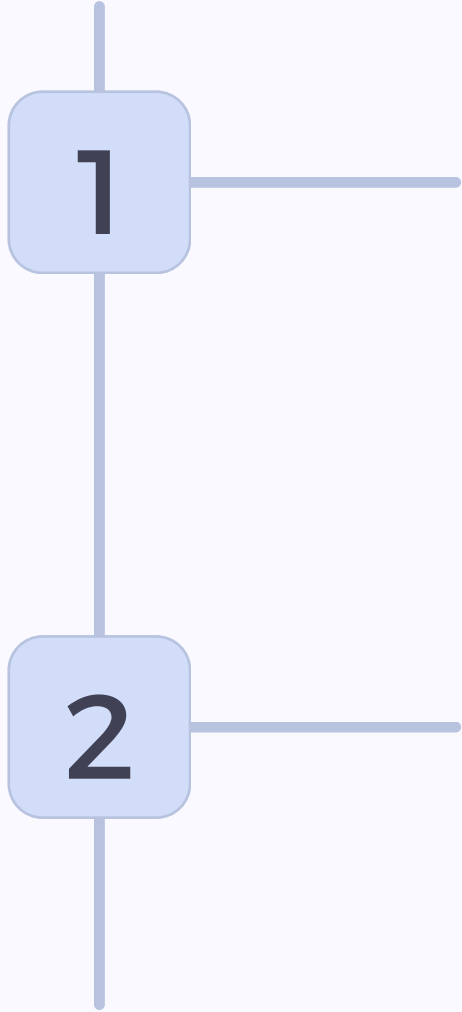
Use `seekp()` and `seekg()` to move the file pointer to a specific position.

`Tellp()` and `Tellg()`



Retrieve the current file pointer position using these functions.

# Error Handling in File Operations

- 
- 1 File Not Found**  
Use **fail()** or **bad()** to check if a file failed to open.
  - 2 Read/Write Errors**  
Handle write or read errors using **fail()**, **bad()**, or **eof()**.

# Best Practices for File Handling



1

## Close files

Always close files when you're done with them.



2

## Use exceptions

Handle file errors gracefully using exceptions.



3

## Flush buffers

Ensure data is written to the file immediately using `flush()`.

# Conclusion

---

You now have a solid understanding of file handling in C++!  
Experiment with different file operations, explore advanced features, and continue to build upon your knowledge to master this essential programming skill.





Thank you