Tutorial 59 - File I/O in C++: Working with Files

Introduction to File Handling in C++

- A **file** is a collection of data stored on a disk.
- Anything written inside a file is called **content** (previously referred to as a "patent").
- A **text file** contains multiple characters, such as `;`, `#include`, etc.
- Computers read text files using **ASCII codes**, where each character is mapped to a decimal number.
  - Example: The ASCII code for 'A' is **65**.
  - These decimal numbers are then converted into **binary (0s and 1s)** since computers only understand **binary language**.
- Computers rely on binary because they are made of **switches**, which operate in two states: **ON (1) and OFF (0)**.

File Input and Output in C++

Files can be used for:

1. **Reading (Input)**
2. **Writing (Output)**

- The user interacts with a C++ program using:
  - `cin >>` → Takes input from the **keyboard**.
  - `cout <<` → Displays output on the **screen**.
- Files allow us to:
  - **Write to a file** (store data).
  - **Read from a file** (retrieve data).

File Read and Write Process

Understanding File I/O Operations

1. **Writing to a File**
   - Stores data permanently in a file.
2. **Reading from a File**
   - Retrieves stored data for use.

📌 File Handling in C++ Uses the fstream Library

```
#include <fstream>

```

Short Notes

Key File Operations in C++

1. **Reading a File** → Extracts data from a file.
2. **Writing to a File** → Saves data into a file.

Important C++ File I/O Functions

- `cin >>` → Takes input from the user.
- `cout <<` → Displays output on the screen.
- `fstream` → Handles file operations in C++.