

Buffering

Writing to console/file is a slow operation

If the program had to write each character **immediately**, the runtime would **significantly slow down**, What can we do?

- Accumulate characters into a buffer
- When buffer is **full**, write out all contents of the buffer to the output device **at once**
- This process is known as **flushing** the stream

Flushing the Buffer

If you want to force the content of the buffer to their destination, we can flush the stream

```
stream.flush();
stream<<std::flush;
```

Not all streams are buffered

- **std::cerr** is not buffered
 - Each time you insert something into it, it flush immediately
- **std::cout** is buffered
 - **std::cout** will only flush if
 - it reaches maximum capacity
 - you explicitly ask it to do so

Stream bits

- **Good bits** No errors, the stream is good to go
- **EOF bits** End-of-file was reached during a previous operation
- **Fail bits** Logical error on a previous operation
- **Bad bit** Likely unrecoverable error on previous operation

Chaining >> or <<

>> and << are not magic, they are actually functions

```
std::cout<<"hello";  $\xrightarrow{\text{is equivalent to}}$  operator<<(std::cout, "hello")
```

- >> is **function**
- std::cout and hello are function’s **arguments**
- >> **return the stream** passed as its left argument
- That’s why we can chain a lot of >>

Stream Manipulator

Useful when you need to format your data

[Check this link](#)