

Programming Languages and Tools: Programming with C++ CS:3210:0003

Lecture/Lab #9

Streams

- A sequence of bytes that can be accessed sequentially
- Over time, may produce/consume unlimited amounts of data



iostream

- Input/output stream library
- 4 standard streams in C++:
 1. cin: tied to the standard input (typically keyboard)
 2. cout: tied to the standard output (typically monitor)
 3. cerr: tied to the standard error (typically monitor), with unbuffered output
 4. clog: tied to the standard error (typically monitor), with buffered output

std::cout - Character Output

- Send data to the console to be printed as text
- <<, the insertion operator
 - Use cout with << to send data to console
 - Polymorphic - works with multiple types
 - Doesn't seem to work with arrays
 - Can print literals or values within variables

std::cout is Buffered

- Outputs to be printed to the console are added to a buffer
- When the buffer is **flushed**, these outputs are printed on the console
- Writing to console is slower than writing to buffer
- For newline, use:
 1. `std::endl`
 2. Character literal `'\n'`
 3. String literal `"\n"`
- `endl` moves to a newline and flushes the output buffer, so literal is usually faster

std::cin - Character Input

- Read input from keyboard using extraction operator >>
 - Input must be stored in a variable
 - Polymorphic
 - Buffered
- We can also input space separated input with cin:

```
int x = 0, y = 0;  
cin >> x >> y;
```

Stream Manipulators

- Used to modify stream
- Invoked by passing to stream
- Some useful stream manipulators/functions:

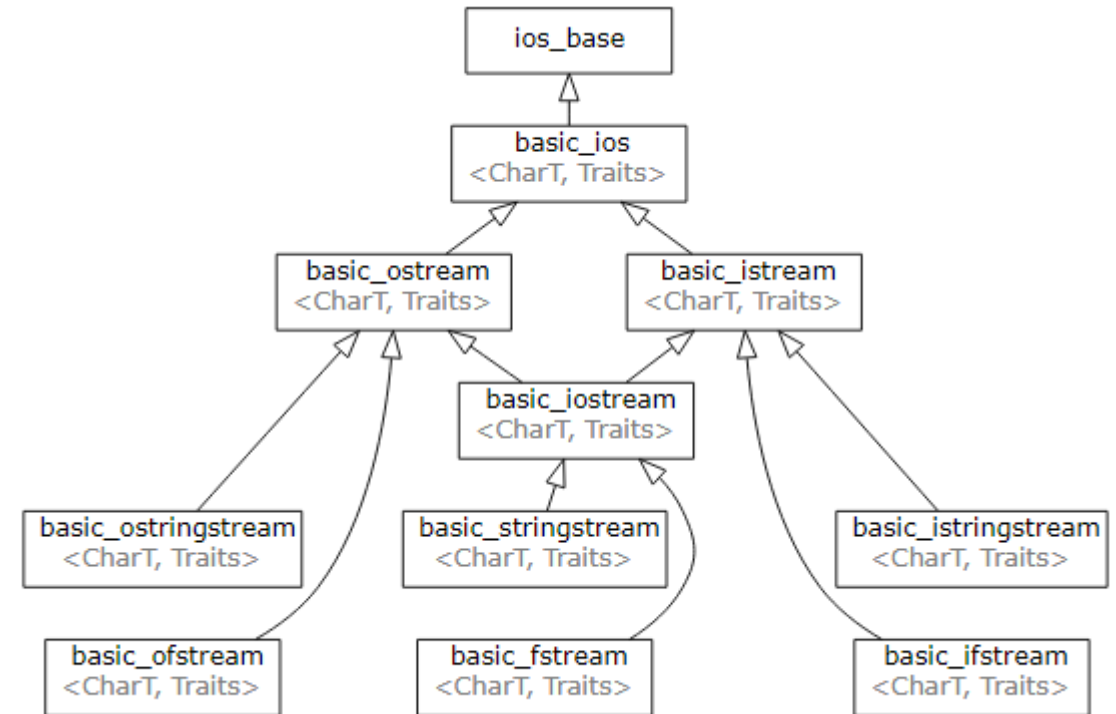
Manipulator	Function
<code>std::endl</code>	Print newline and flush buffered output
<code>std::setw(int n)</code>	Limit no. of chars read from stream
<code>std::ignore()</code>	Discard first char in stream
<code>std::peek()</code>	Read char without removing it from stream
<code>std::unget()</code>	Returns last character read to stream so it can be read again by the next call

I/O Formatting

- Manipulators – objects placed in stream to effect I/O
- **Flags** – bits that can be turned on/off to effect I/O
- Use `setf()` and `unsetf()` to turn flags on/off

C++ Streams

- OOP-style stream-based I/O library
- Implemented within a class hierarchy
- Organized around abstract input/output devices
- Streams for
 - I/O devices
 - Strings
 - Files



Streams for File I/O

- File I/O classes: ifstream, ofstream, and fstream
- Need to set up a file stream corresponding to a file
- Use insertion (<<) and extraction (>>) to read and write from stream
- >> breaks on whitespace
- Use getline() to read by line