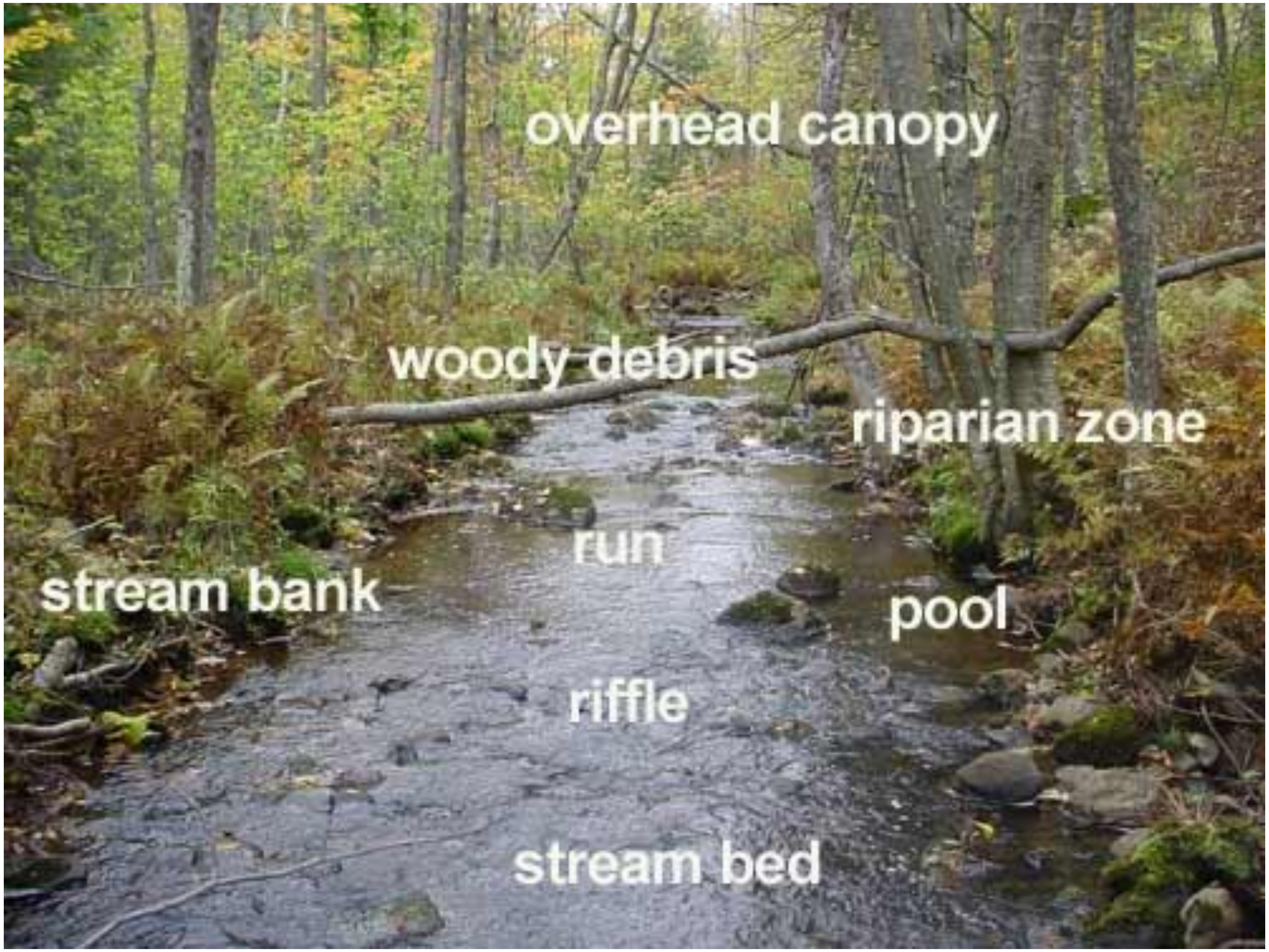
Streams

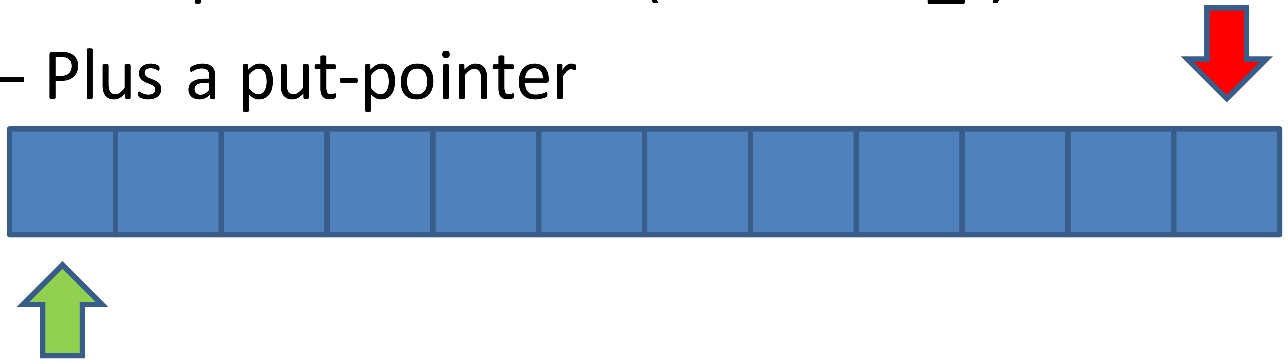
Head << Knowledge

What is a Stream?



What is a Stream?

- For all intents and purposes:
  - A sequence of char (or wchar_t)
  - Plus a put-pointer



  - Plus a get pointer

- No Array! (no random access)

Example

```
// sets the get pointer to the beginning.
seekg(0); seekg(0,ios::beg);
// sets the get pointer to 5 chars forward of the
// beginning.
seekg(5,ios::beg);
// returns the current value of the put/get pointer
tellp(); tellg()
// sets the put pointer to 10 chars before the end
seekp(-10,ios::end);
// proceeds to next char
seekp(1,ios::cur);
```

<< Streaming Operators >>

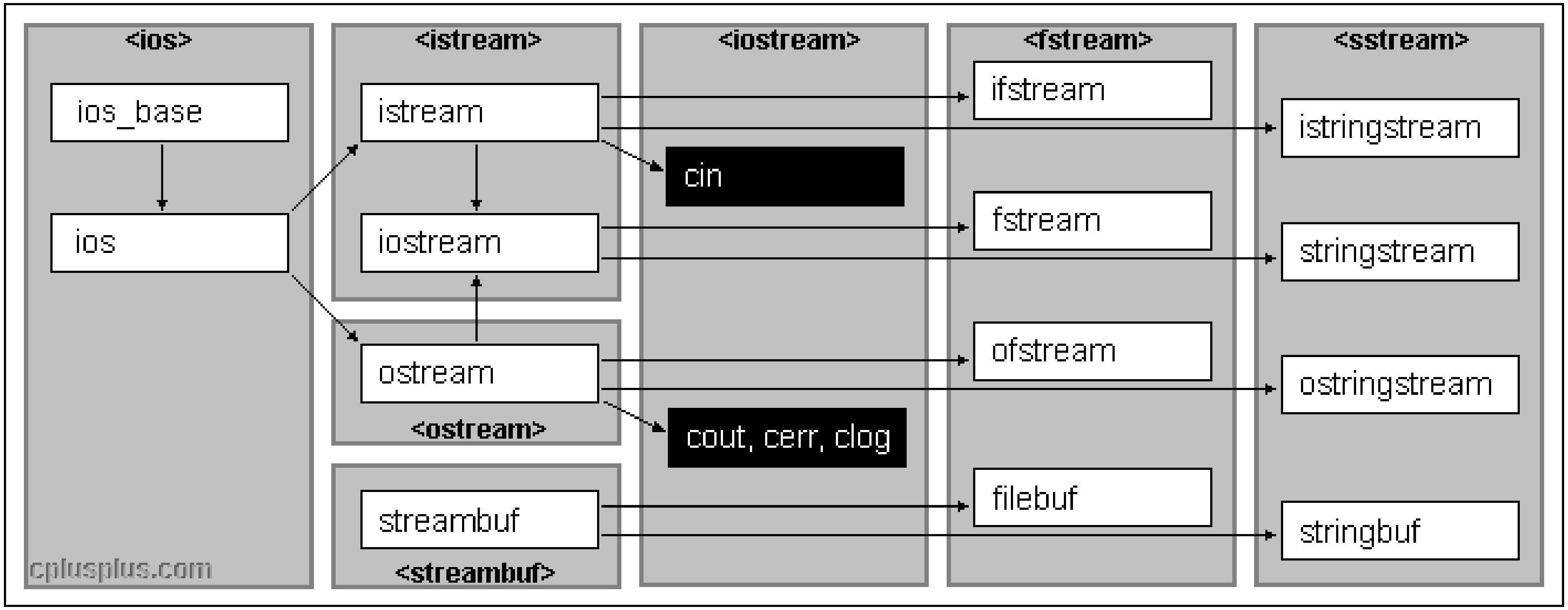
<<

appends an object to a stream

>>

Read object from Stream

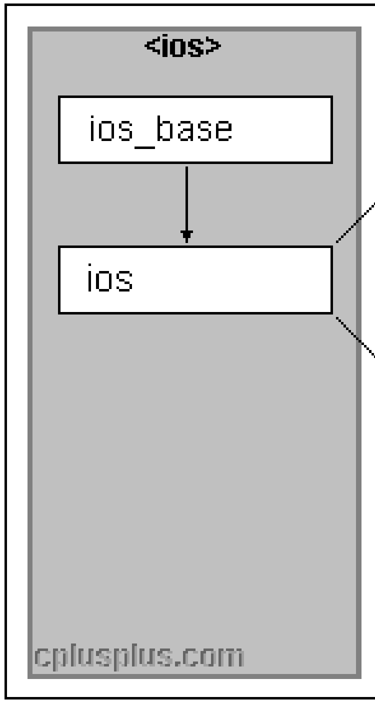
Streams Streams Streams



ios

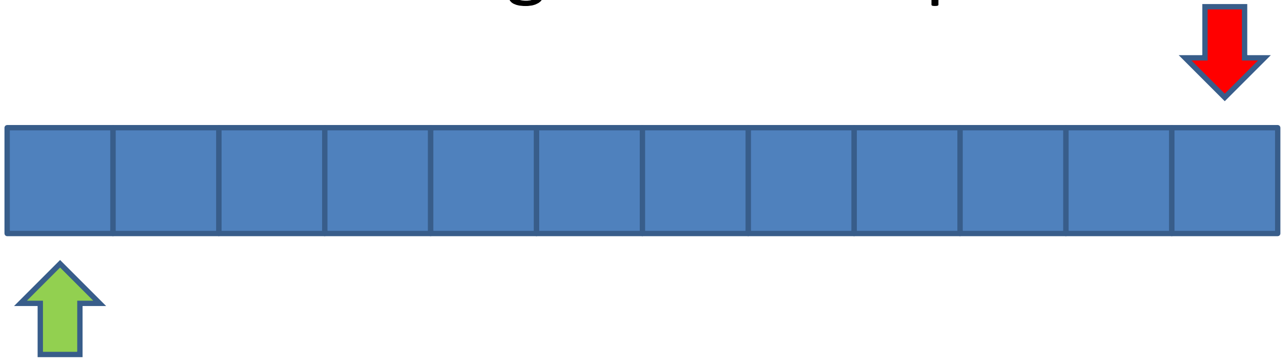
- Basic Stream, in and out
- Independent of in/out

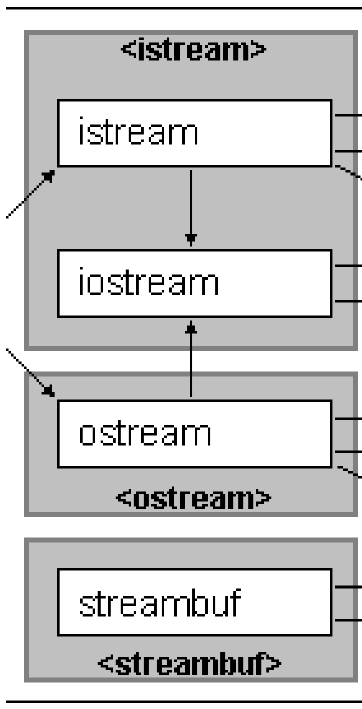
- Rarely used directly



Istream/ostream/streambuf

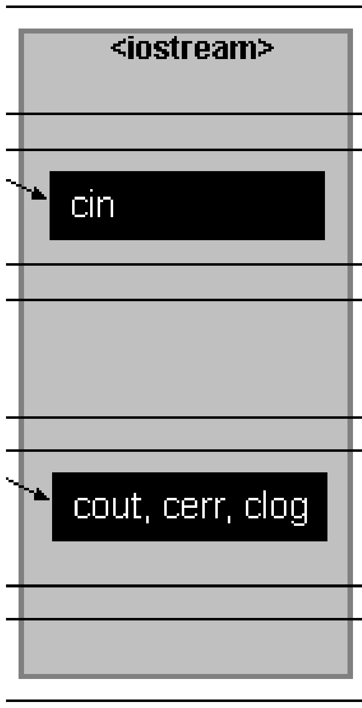
- Specialised versions for each task
- Streambuf: e.g. audio output





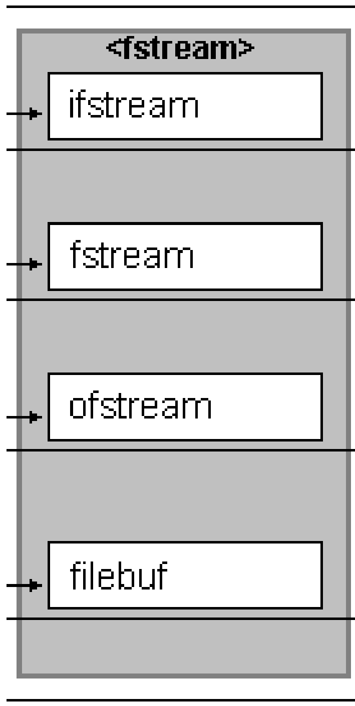
iostream

- Standart streaming inputs and outputs



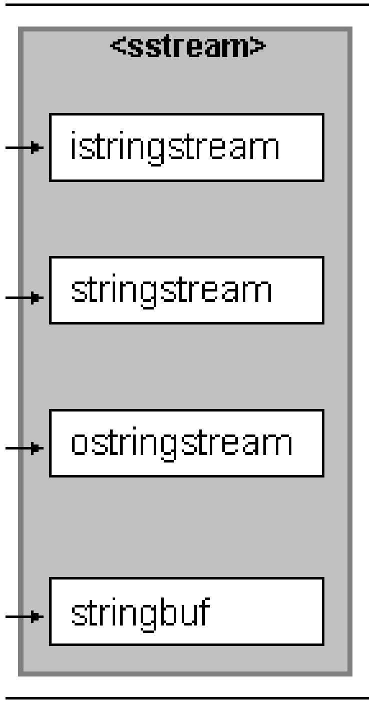
fstream

- Stream to files
- Special tools like eof()…



sstream

- Stream Strings



Why Stream?

- Flexible, simple input/output facility
- Abstract!
- Reusable

Example?

File Reader / - Writer!

Don‘t use Filenames

- Read(std::string filename)

Use Streams!

- Easy reuse in other Reader/Writers
  - e.g. Zip-Files
- Opening a file for continuous input is possible
  - Continuous streaming of data
  - Reserving access

Further Reading

- Introduction, practical examples:
- http://www.cprogramming.com/tutorial/c++-iostreams.html

- Lecture, explains basics and underlying principles:
- http://courses.cs.vt.edu/cs1044/Notes/C04.IO.pdf