12-bit I/O

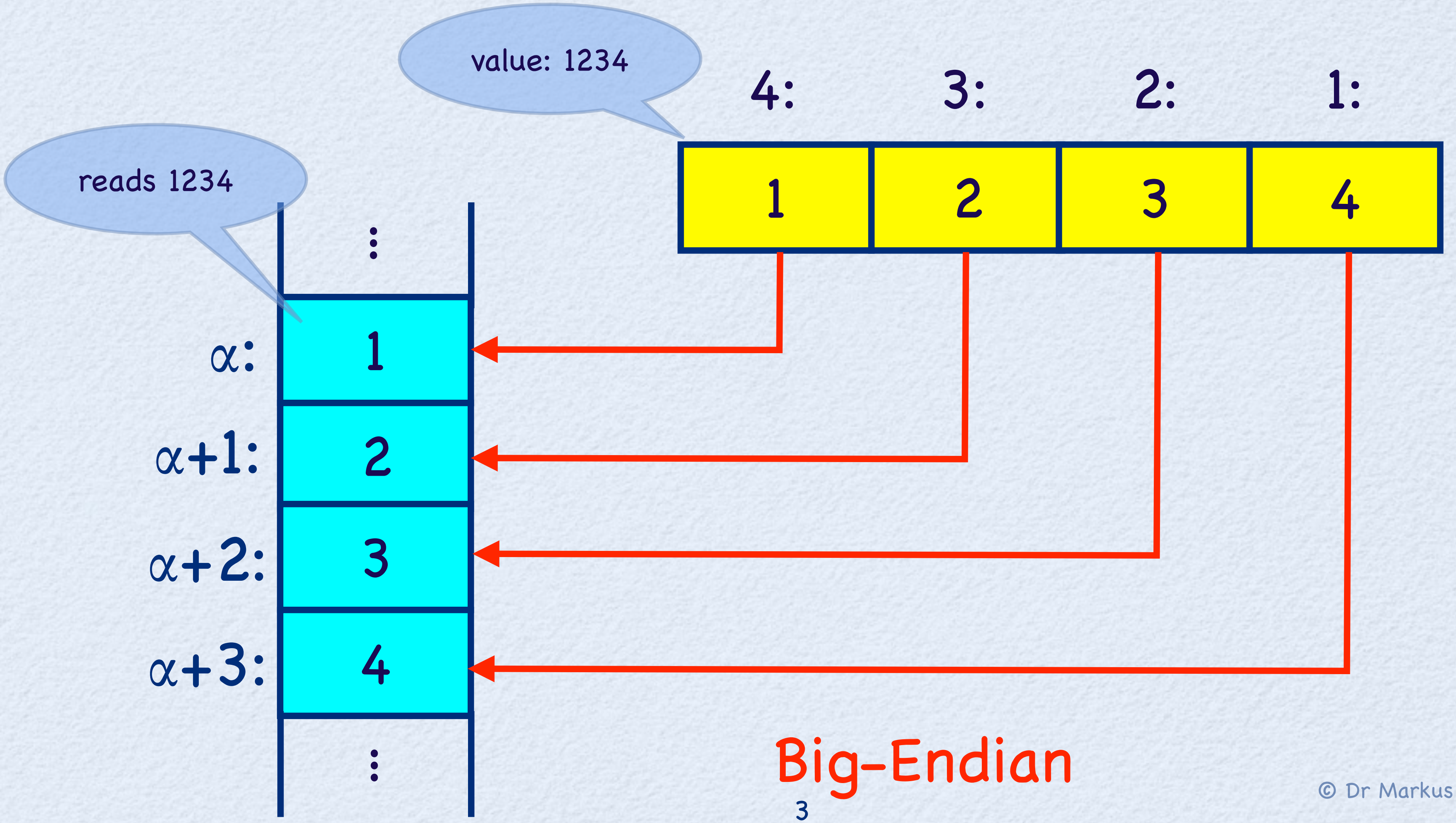
Design Pattern and I/O

Endianness

- When two parties wish to exchange information, then they need to agree on an ordering convention if the data being exchanged is too large to be sent in one piece.
- In computing, endianness refers to the byte or bit ordering of data stored in the computer memory or send over the network.
- We distinguish two orderings:
  - Big-endian order
  - Little-endian order

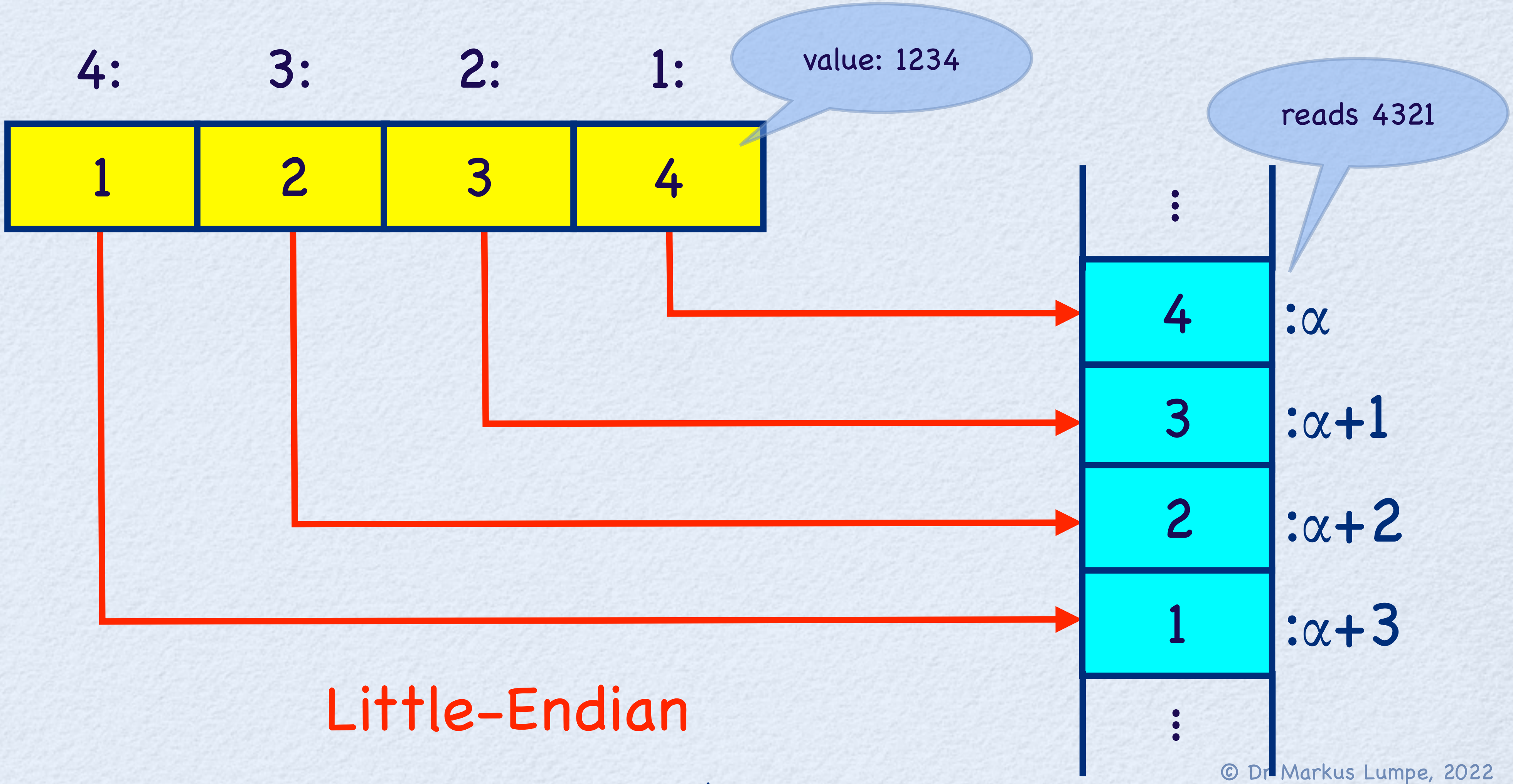
Big-Endian

- The most significant byte or bit (MSB) is stored at the memory location with the lowest address:



Little-Endian

- The least significant byte or bit (LSB) is stored at the memory location with the lowest address:

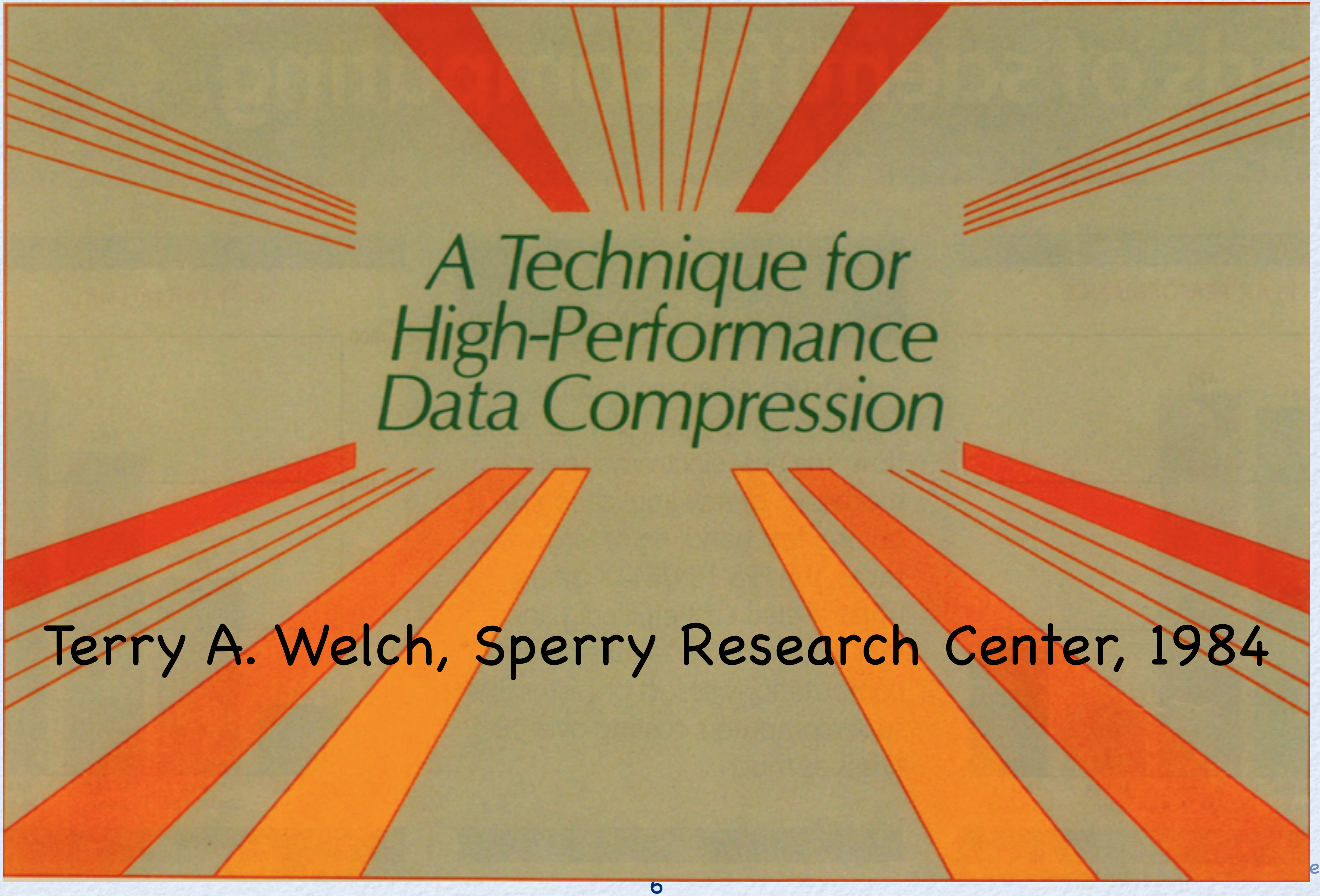


Little-Endian

© Dr. Markus Lumpe, 2022

12-bit values

LZW Compression

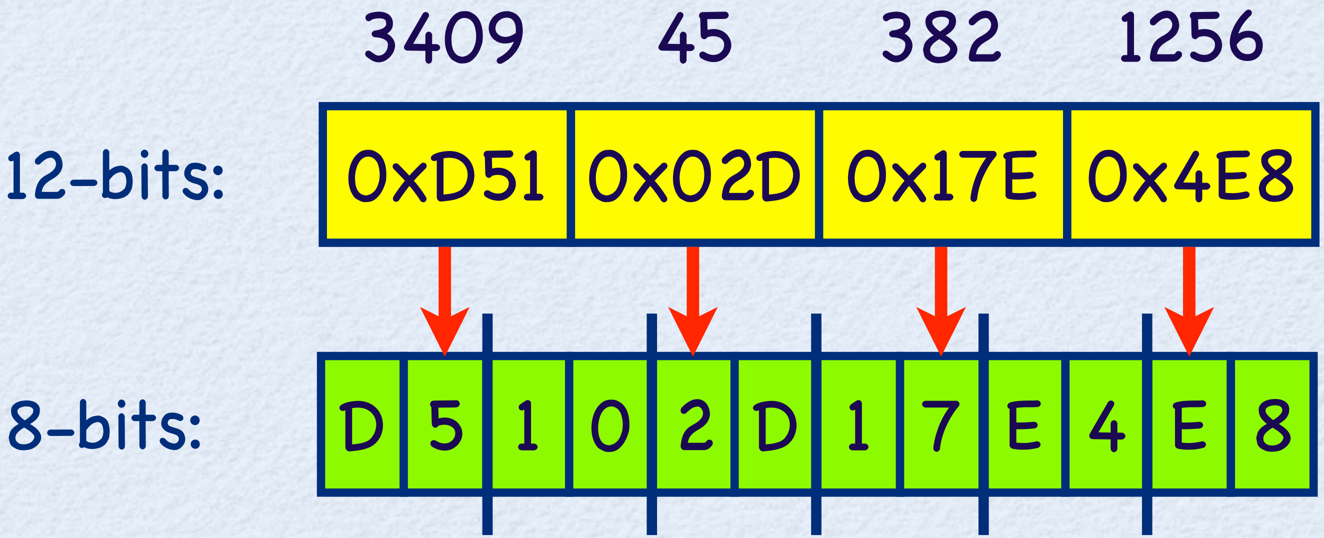


A Technique for High-Performance Data Compression

Terry A. Welch, Sperry Research Center, 1984

Which ordering works best for 12-bit I/O?

- Consider four 12-bit numbers: 1256, 382, 45, and 3409:
  - 12-bit values cannot be stored in a byte.
  - We could use a 16-bit word, but this would waste 4 bits per value.
  - We need to devise an approach where we store the 12-bit values as a consecutive bitstream, each value requiring 12 bits.
  - Which ordering works better: little endian or big endian?

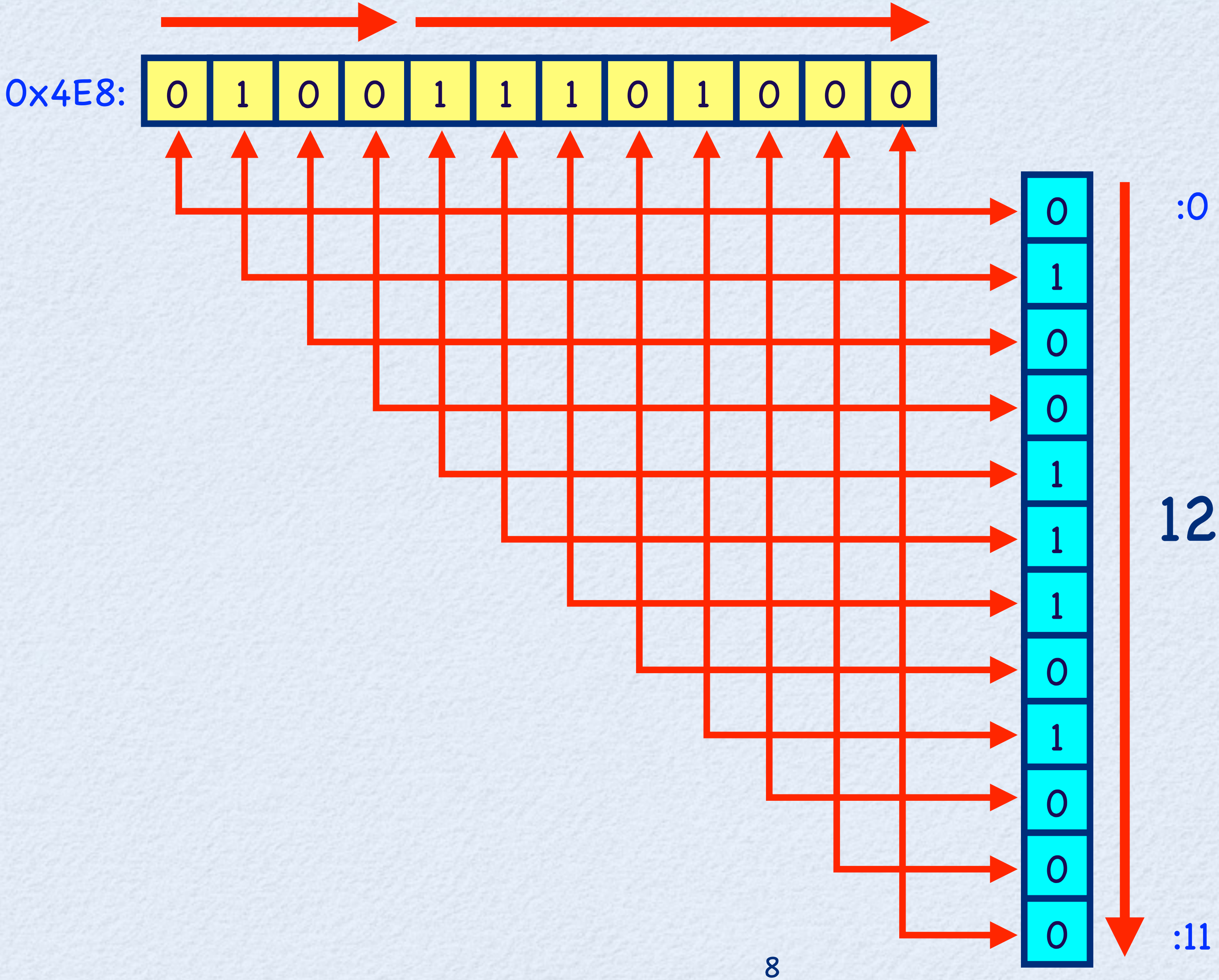


3409 45 382 1256

12-bits: 0xD51 0x02D 0x17E 0x4E8

8-bits: D 5 1 0 2 D 1 7 E 4 E 8

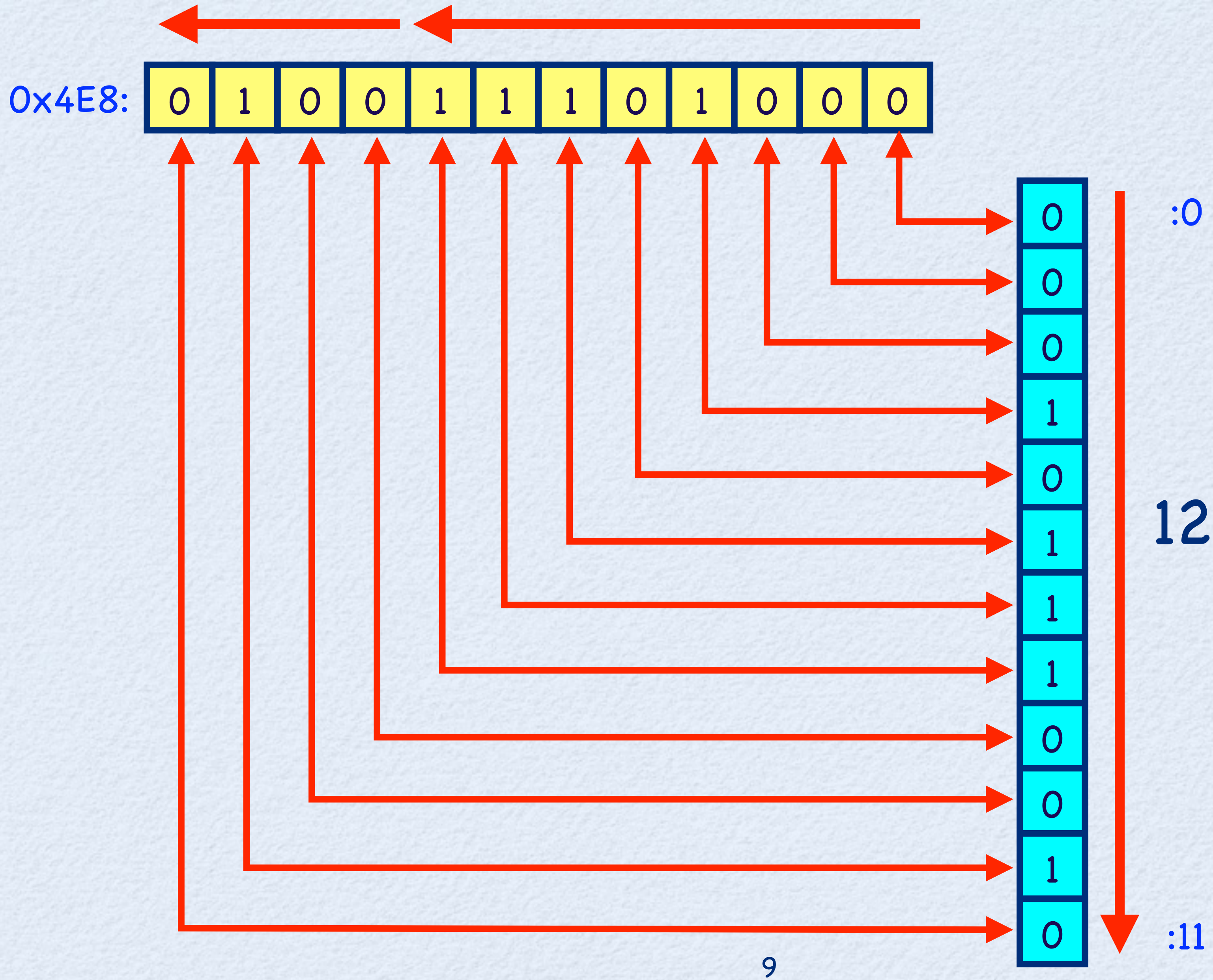
Processing 4E8: Big-Endian



12-bit stream

0x4E8

Processing 4E8: Little-Endian

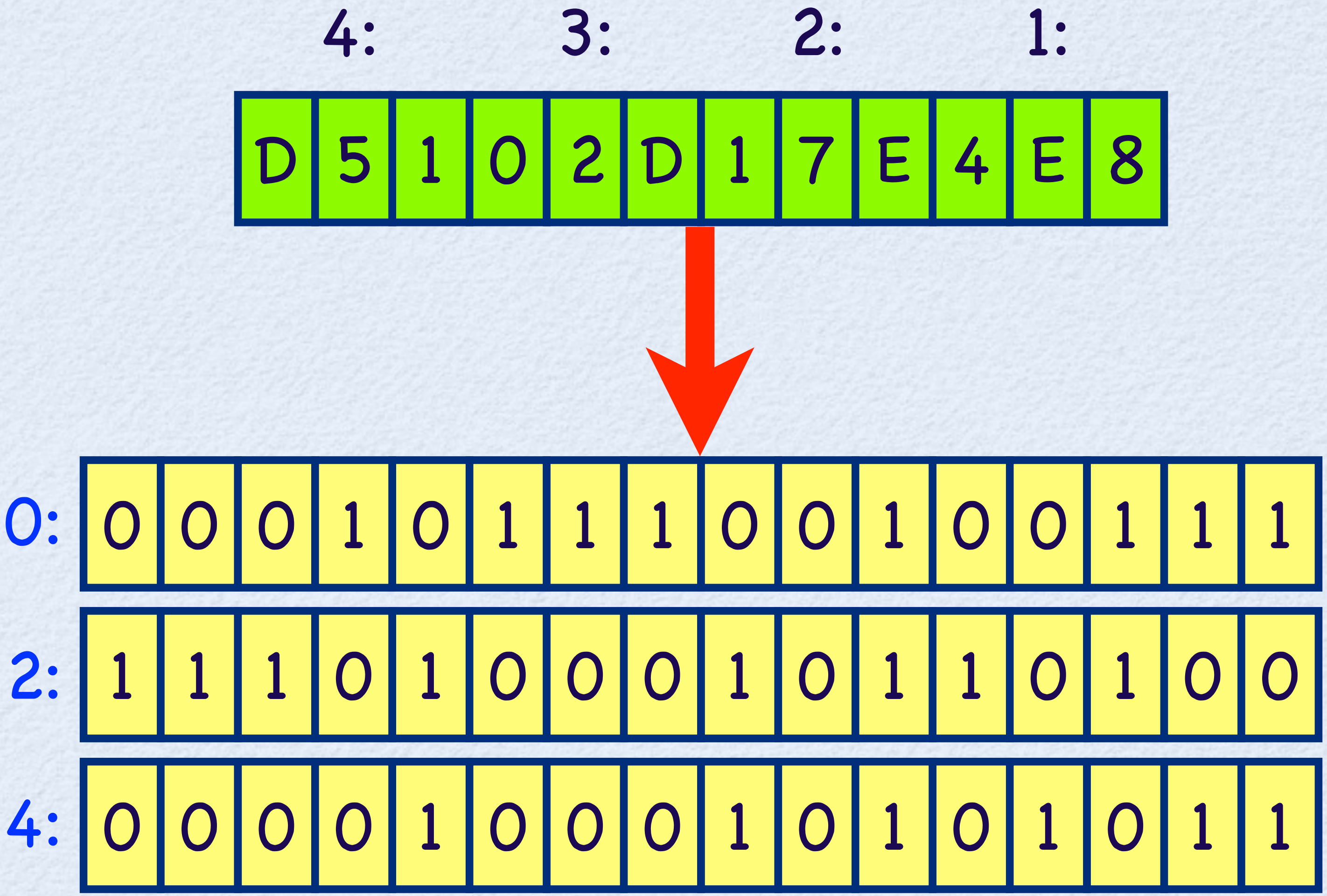


12-bit stream

0x172

4E8 17E 02D D51

- Little-endian bit ordering:



4: 3: 2: 1:

D	5	1	0	2	D	1	7	E	4	E	8

0:	0	0	0	1	0	1	1	1	0	0	1	0	0	1	1	1
2:	1	1	1	0	1	0	0	0	1	0	1	1	0	1	0	0
4:	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	1

Write 12 Bit Values (Pseudocode)

```
write12Bits( aValue : 12Bit ) =
  for i = 1 to 12
    do
      if (aValue & 0x1)          // fetch lowest bit
          then send 1 to output;
          else send 0 to output;
      aValue := aValue / 2;      // divide by 2
    od;
```

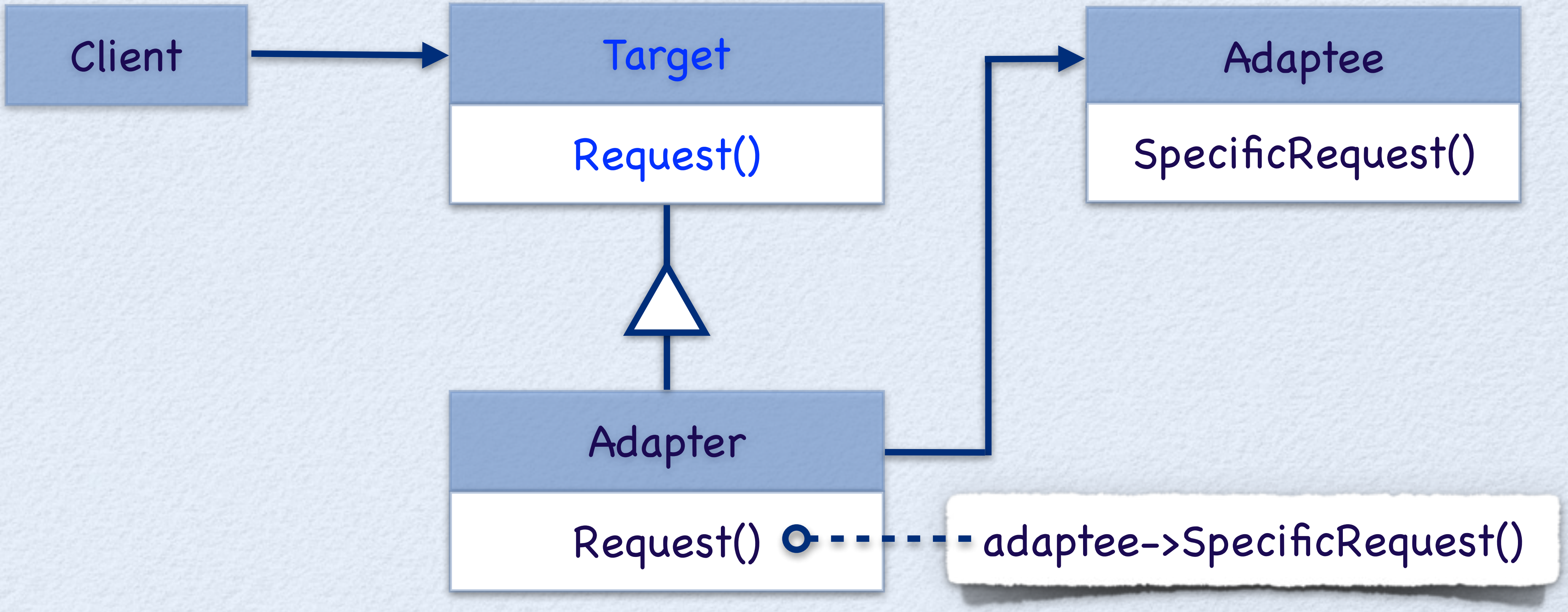
Read 12 Bit Values (Pseudocode)

```
read12Bits() : 12Bit =
   declare Result : 12Bit = 0;
   for i = 1 to 12
     do
       declare lBit : Bit = input()              // get next bit
       if ( lBit == 1 )
         then Result = (1 << (i-1)) + Result;   // set bit at index i
     od;
   return Result;
```

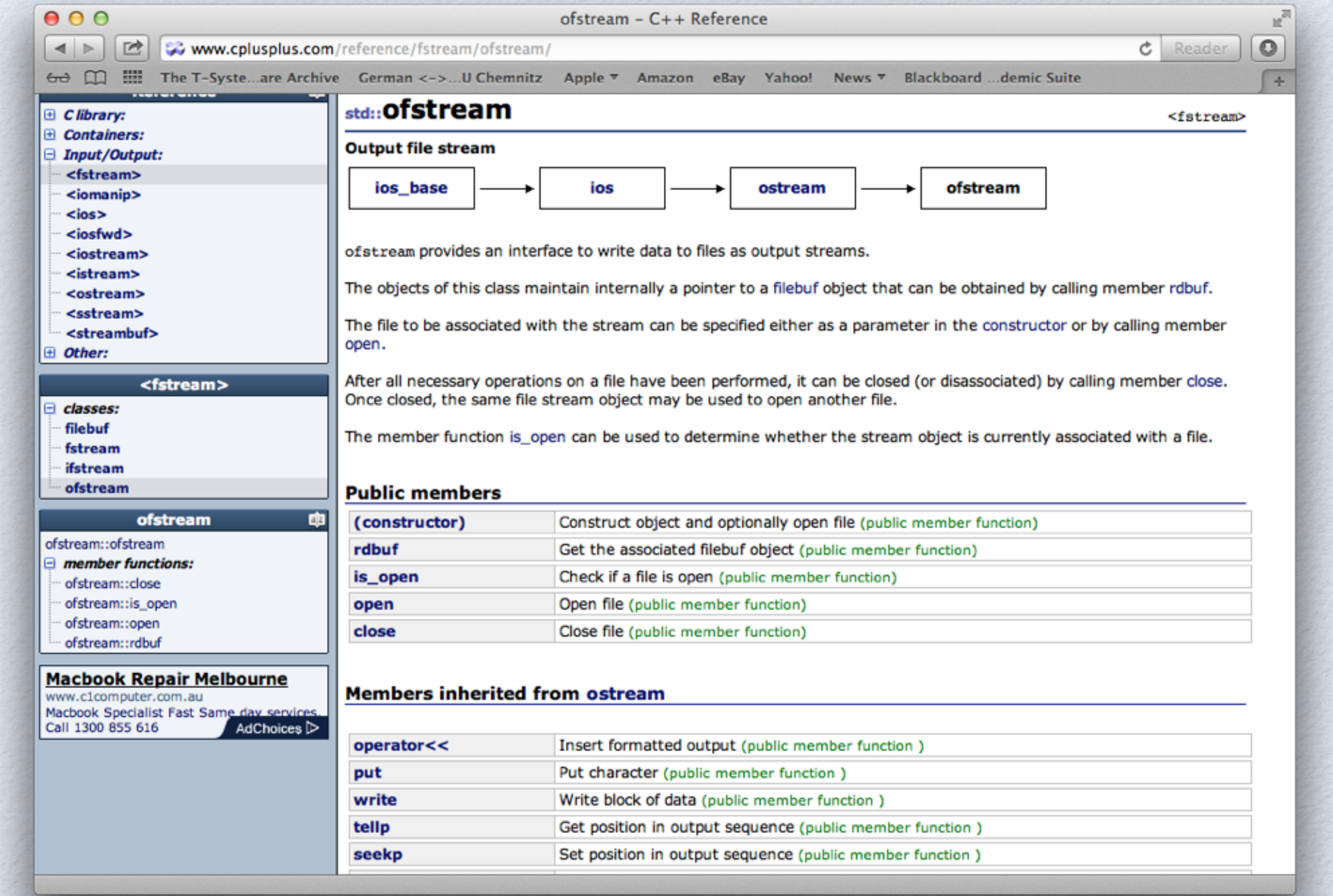
Adapter Design Pattern

- Intent:
  - Convert the interface of a class into another interface clients expect. Adapter lets classes work together that could not otherwise because of incompatible interfaces.
- Collaborations:
  - Clients call operations on an Adapter instance. In turn, the adapter calls Adaptee operations that carry out the request.

Structure of an Object Adapter



Let’s create an object adapter for std::ofstream.



ofstream - C++ Reference
www.cplusplus.com/reference/fstream/ofstream/
Reader
The T-Syste...are Archive
German <->...U Chemnitz
Apple
Amazon
eBay
Yahoo!
News
Blackboard ...demic Suite

⊞ C library:
⊞ Containers:
⊟ Input/Output:
- <fstream>
- <iomanip>
- <ios>
- <iosfwd>
- <iostream>
- <istream>
- <ostream>
- <sstream>
- <streambuf>

⊞ Other:

<fstream>

⊟ classes:
- filebuf
- fstream
- ifstream
- ofstream

ofstream

ofstream::ofstream

⊟ member functions:
- ofstream::close
- ofstream::is_open
- ofstream::open
- ofstream::rdbuf

std::ofstream

<fstream>

Output file stream

ios_base → ios → ostream → ofstream

`ofstream` provides an interface to write data to files as output streams.

The objects of this class maintain internally a pointer to a filebuf object that can be obtained by calling member rdbuf.

The file to be associated with the stream can be specified either as a parameter in the constructor or by calling member open.

After all necessary operations on a file have been performed, it can be closed (or disassociated) by calling member close. Once closed, the same file stream object may be used to open another file.

The member function is_open can be used to determine whether the stream object is currently associated with a file.

Public members

(constructor)	Construct object and optionally open file (public member function)
rdbuf	Get the associated filebuf object (public member function)
is_open	Check if a file is open (public member function)
open	Open file (public member function)
close	Close file (public member function)

Members inherited from ostream

operator<<	Insert formatted output (public member function)
put	Put character (public member function)
write	Write block of data (public member function)
tellp	Get position in output sequence (public member function)
seekp	Set position in output sequence (public member function)