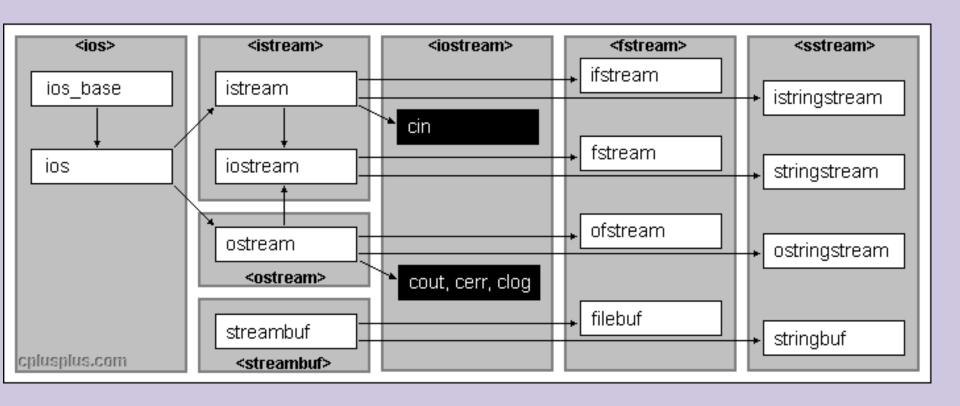
More on Streams

Version 1: Dr. Ofir Pele

Version 2: Dr. Erel Segal-Halevi

Class hierarchy



Standard output stream objects (folder 1)

- cout attached to stdout, buffered.
- cerr attached to stderr, unbuffered.
- clog attached to stderr, may be buffered or unbuffered.

We can redirect stdout and stderr to different files.

Output stream manipulators (folder 3)

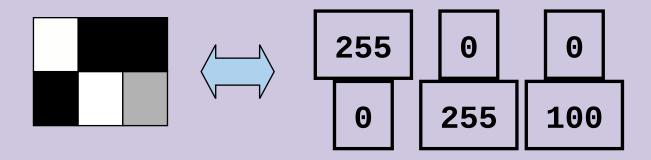
- We can "write" to ostream, functions that do not create any output, but rather change some variables of the ostream.
- For example:
- "setprecision" does not print anything it just modifies the precision level of the stream.
- How does it work? operator overloading!

http://cs.brown.edu/~jwicks/libstdc++/html_user/iomanip-source.html

Binary files

Leading example: image files

- Images are stored as matrices of numbers (pixels)
- Here, we deal with gray-scale images
- 8 bits per pixel
 - i.e. each pixel between 0 and 255
- 255 is white, 0 is black, others are gray



storing images

- How can we store images on files?
- For each image we want to store:
 - width
 - height
 - number of bytes per pixel
 - the pixels
- Requirements: read/write easily, save space, save computation, etc.

storing images

First try: text files

cons:

- long
- needs parsing

pros:

- readable by humans
- easy to edit

"myImg.txt"

```
width = 3
height = 2
bytes_per_pixel = 1
255 0 0
0 255 100
```

storing images

Better solution: Binary files

Save the data the way the computer holds it

pros:

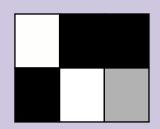
- Smaller
- No parsing (faster)

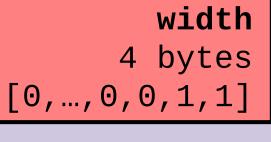
cons:

- hard to read for humans
- Machine dependant

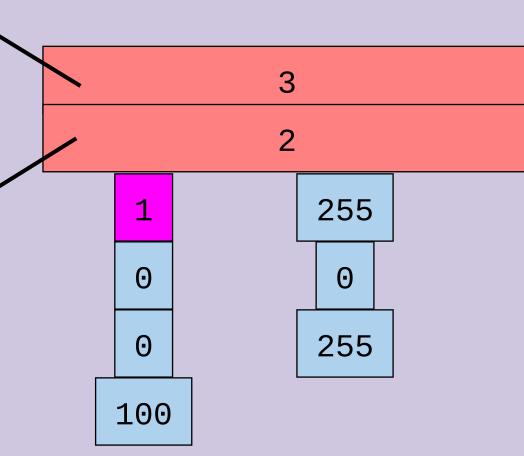
Widely used:
 JPEG, mp3, BMP, other data

Images as binary files

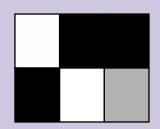


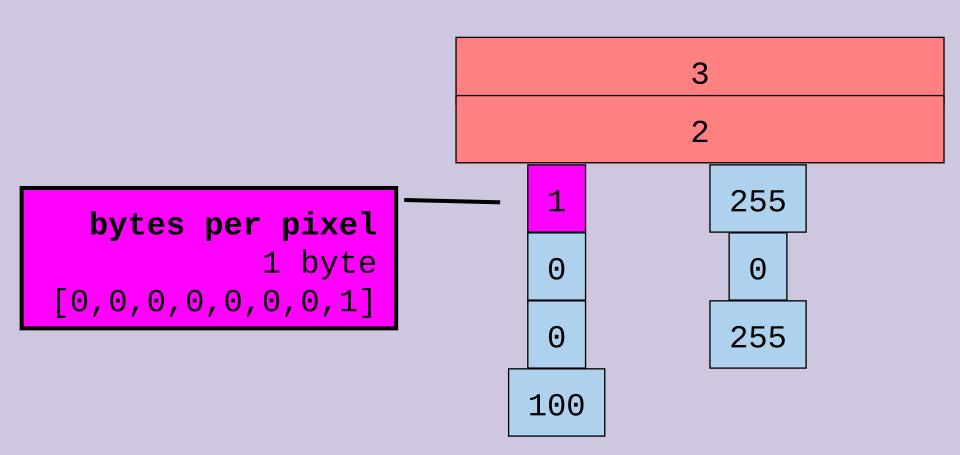


height
4 bytes
[0,...,0,0,1,0]

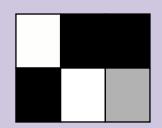


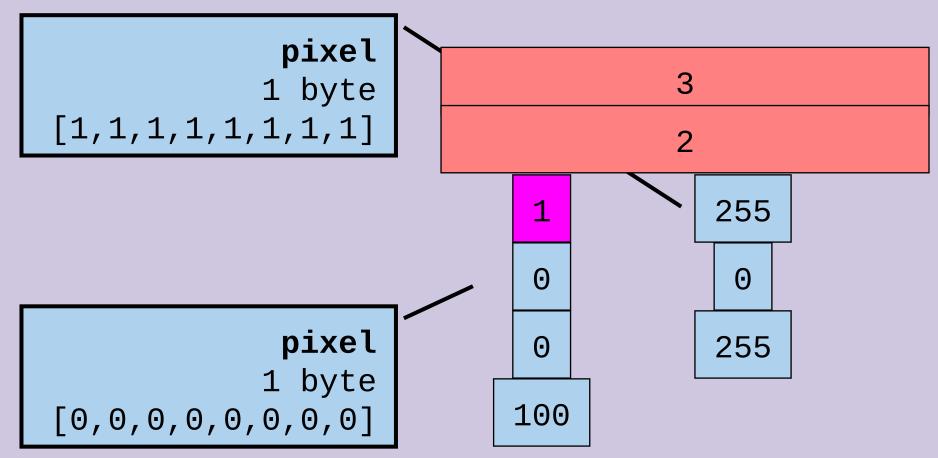
Images as binary files





Images as binary files





Images as binary files - colors

In a colorful image, each pixel should contain more information than just the light intensity.

A common way to represent colors is RGB (Red, Green, Blue).

Each pixel requires 3 bytes – one for Red, one for Green, one for Blue.

See example in folder 5.