C++ Output Formatting (iostream) — Cheat Sheet

Purpose:

This cheat sheet summarizes commonly used output manipulators, flags, and formatting options in C++. It covers line breaks, buffering, width, padding, alignment, floating-point formatting, integer bases, booleans, strings, generic flag control, and locale-aware time/money formatting. Examples use `std::cout`

Tip:

1. `setw (width)` applies to the next insertion only.
2. Most other flags persist until changed.

Required headers:

```
#include <iostream>
#include <iomanip>
#include <ios>
#include <string>
#include <ctime> (for std::tm and put_time)
#include <locale> (for put_money)
```

1) Line breaks and buffering

Manipulator	Header	What it does / Example
`endl`	<ostream>	`cout << "hi" << endl;`
`\n`	n/a	`cout << "hi\n";`
`flush`	<ostream>	`cout << flush;`
`ends`	<ostream>	`cout << '\0';`
`unitbuf`	<ios>	`cout << unitbuf;`
`nounitbuf`	<ios>	`cout << nounitbuf;`

2) Width, padding, alignment

Manipulator	Header	What it does / Example
`setw(n)`	<iomanip>	`cout << setw(6) << 42;`
`setfill(c)`	<iomanip>	`cout << setfill('-');`
`left`	<iomanip>	`cout << left;`
`right`	<iomanip>	`cout << right;`
`internal`	<iomanip>	`cout << internal;`

3) Floating-point formatting

Manipulator	Header	What it does / Example
`setprecision(n)`	<iomanip>	`cout << setprecision(4);`
`fixed`	<ios>	`cout << fixed;`
`scientific`	<ios>	`cout << scientific;`
`defaultfloat`	<ios>	`cout << defaultfloat;`
`hexfloat`	<ios>	`cout << hexfloat;`
`showpoint`	<ios>	`cout << showpoint;`
`noshowpoint`	<ios>	`cout << noshowpoint;`

4) Integer base and number style

Manipulator	Header	What it does / Example
`dec`	<ios>	`cout << dec;`
`oct`	<ios>	`cout << oct;`
`hex`	<ios>	`cout << hex;`
`setbase(n)`	<iomanip>	`cout << setbase(16);`
`showbase`	<ios>	`cout << showbase;`
`noshowbase`	<ios>	`cout << noshowbase;`
`uppercase`	<ios>	`cout << uppercase;`
`nouppercase`	<ios>	`cout << nouppercase;`
`showpos`	<ios>	`cout << showpos;`
`noshowpos`	<ios>	`cout << noshowpos;`

5) Booleans and strings

Manipulator	Header	What it does / Example
`boolalpha`	<ios>	`cout << boolalpha;`
`noboolalpha`	<ios>	`cout << noboolalpha;`
`quoted(s)`	<iomanip>	`cout << quoted("a b");`

6) Generic flag control

Manipulator	Header	What it does / Example
`setiosflags(f)`	<iomanip>	`cout.setf(ios::showpos);`
`resetiosflags(f)`	<iomanip>	`cout.unsetf(ios::showpos);`

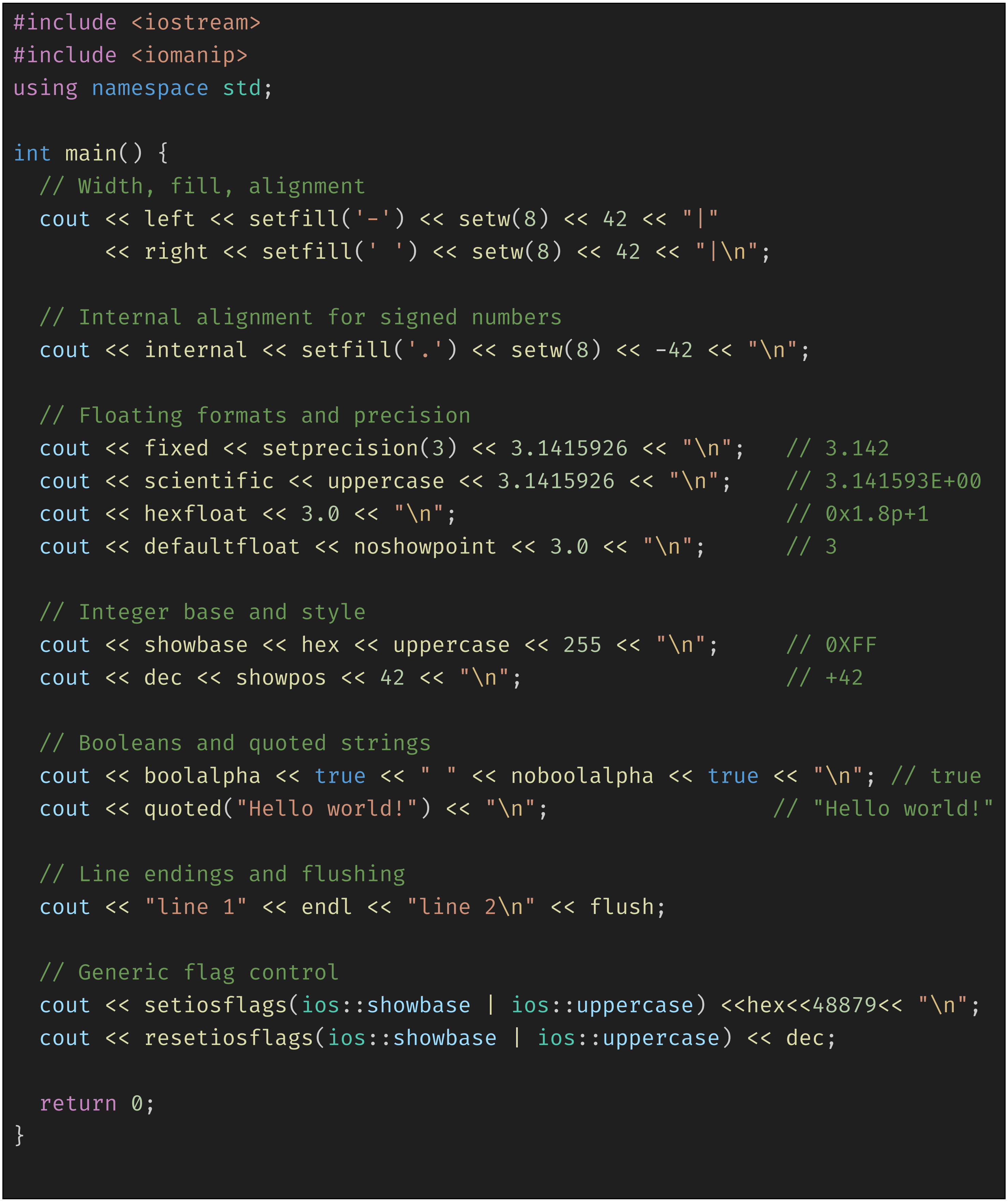
Common flags for setiosflags / resetiosflags:

```
ios::left, ios::right, ios::internal, ios::dec, ios::oct, ios::hex, ios::fixed,
ios::scientific, ios::boolalpha, ios::showbase, ios::showpoint, ios::showpos,
ios::uppercase, ios::unitbuf
```

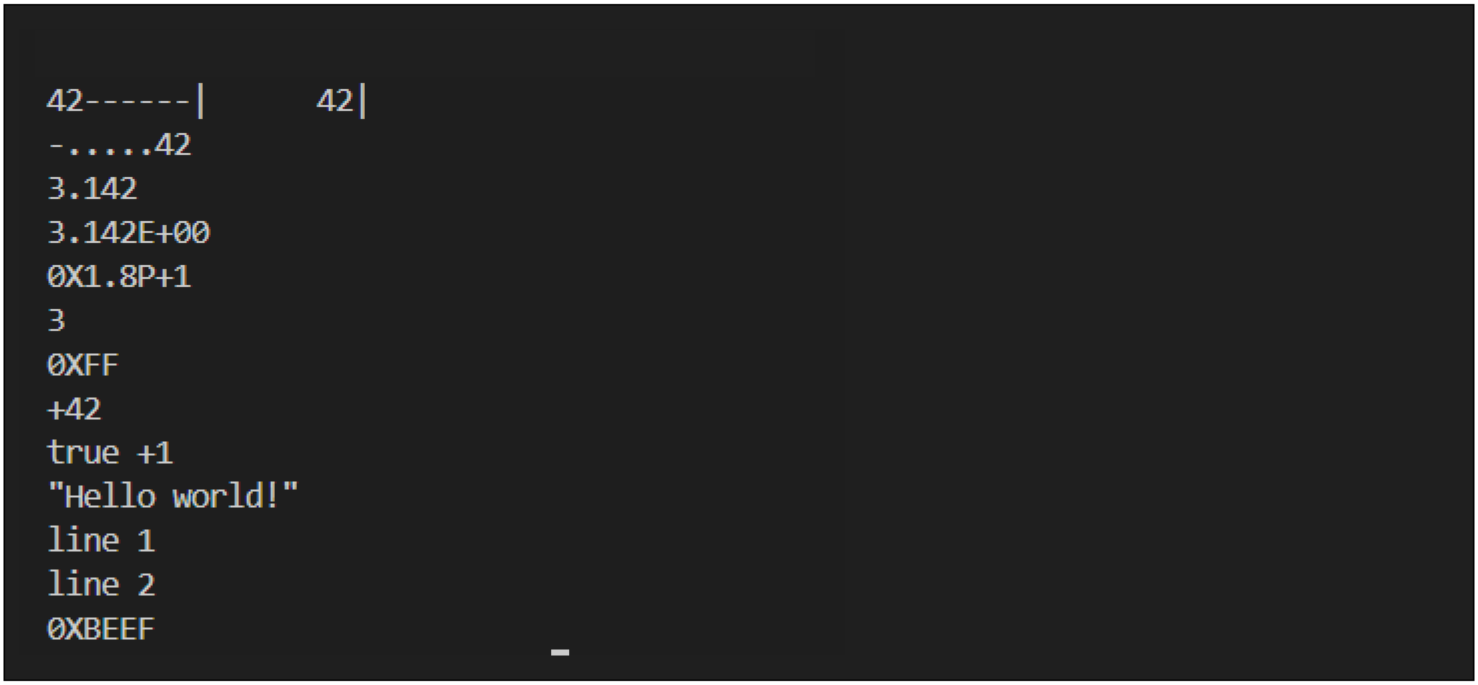
7) Time and money (locale-aware)

Manipulator	Header	What it does / Example
`put_time`	<iomanip>	`cout << put_time(&myTime, "%Y-%m-%d%H:%M");`
`put_money`	<iomanip>	`cout << put_money(amount);`

8) Quick Examples



```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
  // Width, fill, alignment
  cout << left << setfill('-') << setw(8) << 42 << "|"
       << right << setfill(' ') << setw(8) << 42 << "|\n";

  // Internal alignment for signed numbers
  cout << internal << setfill('.') << setw(8) << -42 << "\n";

  // Floating formats and precision
  cout << fixed << setprecision(3) << 3.1415926 << "\n";   // 3.142
  cout << scientific << uppercase << 3.1415926 << "\n";    // 3.141593E+00
  cout << hexfloat << 3.0 << "\n";                         // 0x1.8p+1
  cout << defaultfloat << noshowpoint << 3.0 << "\n";      // 3

  // Integer base and style
  cout << showbase << hex << uppercase << 255 << "\n";     // 0XFF
  cout << dec << showpos << 42 << "\n";                    // +42

  // Booleans and quoted strings
  cout << boolalpha << true << " " << noboolalpha << true << "\n"; // true
  cout << quoted("Hello world!") << "\n";                 // "Hello world!"

  // Line endings and flushing
  cout << "line 1" << endl << "line 2\n" << flush;

  // Generic flag control
  cout << setiosflags(ios::showbase | ios::uppercase) <<hex<<48879<< "\n";
  cout << resetiosflags(ios::showbase | ios::uppercase) << dec;

  return 0;
}
```



```
42------|      42|
-.....42
3.142
3.142E+00
0X1.8P+1
3
0XFF
+42
true +1
"Hello world!"
line 1
line 2
0XBEEF
```

Notes and teaching tips

- **Persistence**: `setprecision, fixed, hex, showbase, etc. persist until changed; setw` applies to the next insertion only.
- **fixed vs scientific**: With `fixed, setprecision(n)` is digits *after* the decimal; otherwise it's significant digits.
- **uppercase** affects hexadecimal digits and the exponent letter in scientific notation.
- **showbase** works with `oct (0)` and `hex (0x/0X);` it has no effect on `dec`.
- **ends** inserts a null character; useful with `std::ostringstream` → C-style buffers.
- **Locale**: `put_money` and `put_time` use the stream's locale (`imbue` `a std::locale` to change).