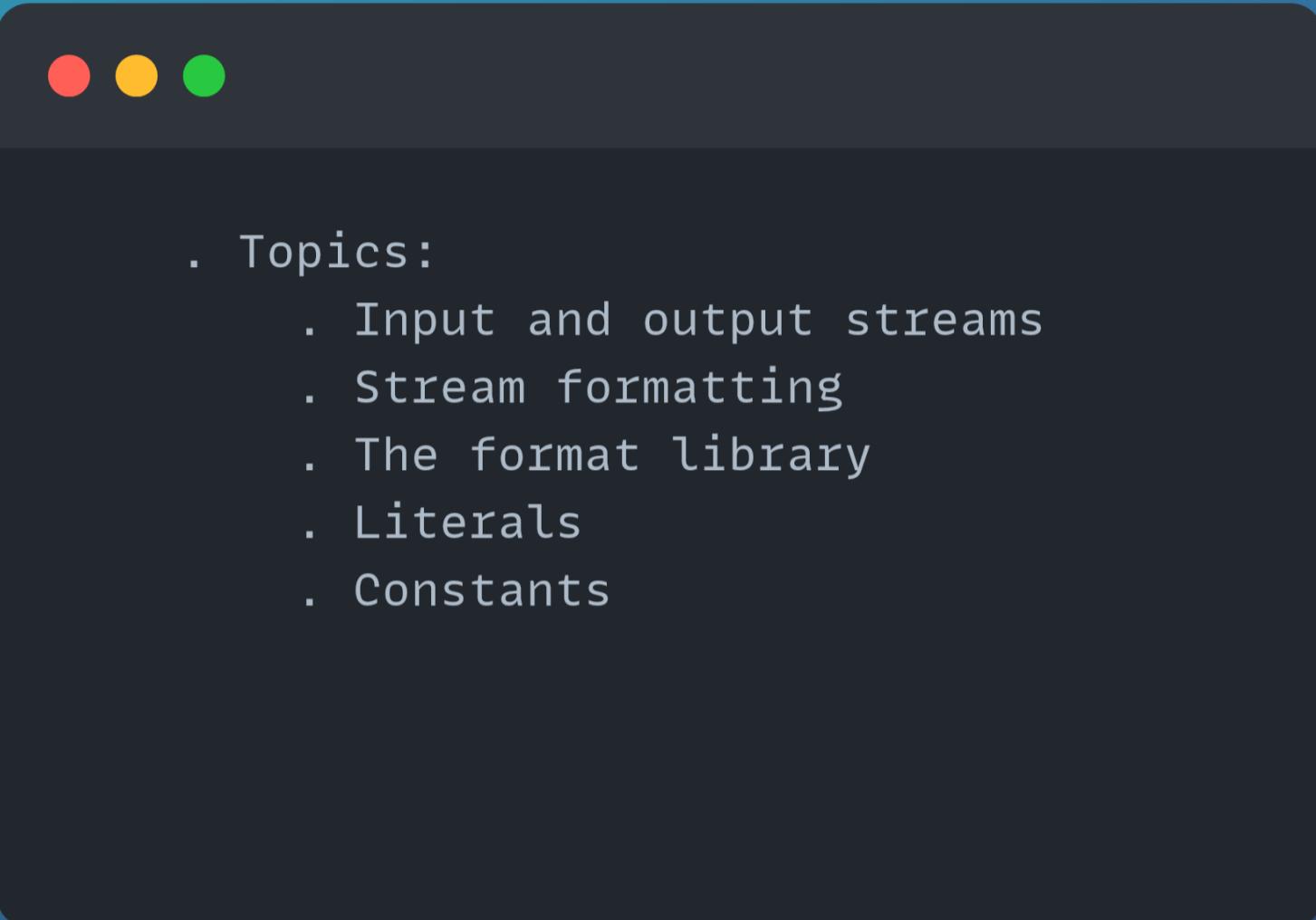
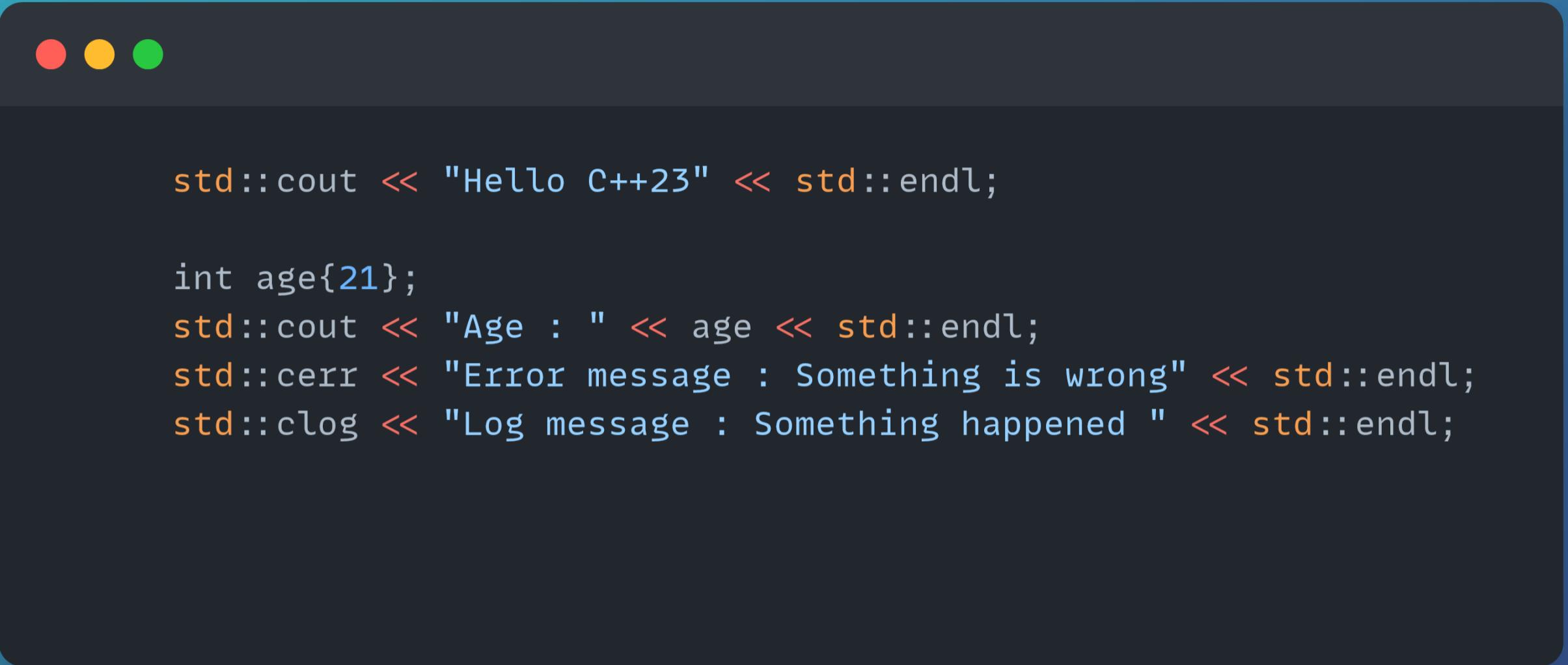


Streams, Formatting, Literals and Constants



- . Topics:
 - . Input and output streams
 - . Stream formatting
 - . The format library
 - . Literals
 - . Constants

iostream



```
std::cout << "Hello C++23" << std::endl;

int age{21};
std::cout << "Age : " << age << std::endl;
std::cerr << "Error message : Something is wrong" << std::endl;
std::clog << "Log message : Something happened " << std::endl;
```

iostream

```
// Data input
int age1;
std::string name;

std::cout << "Please type your name and age : " << std::endl;

// std::cin >> name;
// std::cin >> age1;

std::cin >> name >> age1;

std::cout << "Hello " << name << ". You are " << age1 << " years old!" << std::endl;
```

iostream



```
// Data with spaces

std::string full_name;
int age3;

std::cout << "Please type in your full name and age " << std::endl;
std::getline(std::cin, full_name);

std::cin >> age3;

std::cout << "Hello " << full_name << ". You are " << age3 << " years old!" << std::endl;
```

iostream



```
// Data with spaces

std::string full_name;
int age3;

std::cout << "Please type in your full name and age " << std::endl;

std::getline(std::cin, full_name);

std::cin >> age3;

std::cout << "Hello " << full_name << ". You are " << age3 << " years old!" << std::endl;
```

iostream

```
//Output formatting
// std::endl : places a new line character on the output stream.
//             This is identical to placing '\n' on the output stream.

std::cout << "Hello";
std::cout << "World";

std::cout << std::endl;

std::cout << "-----" << std::endl;

std::cout << "Hello" << std::endl;
std::cout << "World" << std::endl;

std::cout << std::endl;

std::cout << "Hello\n";
std::cout << "World\n";
```

iostream

```
// std::setw() : Adjusts the field width for the item about to be printed.  
// The setw() manipulator only affects the next value to be printed.  
  
std::cout << "Unformatted table : " << std::endl;  
std::cout << "Daniel" << " " << "Gray" << " 25" << std::endl;  
std::cout << "Stanley" << " " << "Woods" << " 33" << std::endl;  
std::cout << "Jordan" << " " << "Parker" << " 45" << std::endl;  
std::cout << "Joe" << " " << "Ball" << " 21" << std::endl;  
std::cout << "Josh" << " " << "Carr" << " 27" << std::endl;  
std::cout << "Izaiah" << " " << "Robinson" << " 29" << std::endl;  
  
std::cout << std::endl;  
std::cout << "Formatted table : " << std::endl;  
std::cout << std::setw(10) << "Lastname" << std::setw(10) << "Firstname" << std::setw(5) << "Age" << std::endl;  
std::cout << std::setw(10) << "Daniel" << std::setw(10) << "Gray" << std::setw(5) << "25" << std::endl;  
std::cout << std::setw(10) << "Stanley" << std::setw(10) << "Woods" << std::setw(5) << "33" << std::endl;  
std::cout << std::setw(10) << "Jordan" << std::setw(10) << "Parker" << std::setw(5) << "45" << std::endl;  
std::cout << std::setw(10) << "Joe" << std::setw(10) << "Ball" << std::setw(5) << "21" << std::endl;  
std::cout << std::setw(10) << "Josh" << std::setw(10) << "Carr" << std::setw(5) << "27" << std::endl;  
std::cout << std::setw(10) << "Izaiah" << std::setw(10) << "Robinson" << std::setw(5) << "29" << std::endl;
```

Unformatted table :
Daniel Gray 25
Stanley Woods 33
Jordan Parker 45
Joe Ball 21
Josh Carr 27
Izaiah Robinson 29

Formatted table :

Lastname	Firstname	Age
Daniel	Gray	25
Stanley	Woods	33
Jordan	Parker	45
Joe	Ball	21
Josh	Carr	27
Izaiah	Robinson	29

iostream

```
// right justified
std::cout << std::endl;
std::cout << "Right justified table (default) : " << std::endl;
col_width = 20;
std::cout << std::right; // Right justify
std::cout << std::setw(col_width) << "Lastname" <<
    std::setw(col_width) << "Firstname" <<
    std::setw(col_width / 2) << "Age" << std::endl;
std::cout << std::setw(col_width) << "Daniel" <<
    std::setw(col_width) << "Gray" <<
    std::setw(col_width / 2) << "25" << std::endl;
std::cout << std::setw(col_width) << "Stanley" <<
    std::setw(col_width) << "Woods" <<
    std::setw(col_width / 2) << "33" << std::endl;
std::cout << std::setw(col_width) << "Jordan" <<
    std::setw(col_width) << "Parker" <<
    std::setw(col_width / 2) << "45" << std::endl;
std::cout << std::setw(col_width) << "Joe" <<
    std::setw(col_width) << "Ball" <<
    std::setw(col_width / 2) << "21" << std::endl;
std::cout << std::setw(col_width) << "Josh" <<
    std::setw(col_width) << "Carr" <<
    std::setw(col_width / 2) << "27" << std::endl;
std::cout << std::setw(col_width) << "Izaiah" <<
    std::setw(col_width) << "Robinson" <<
    std::setw(col_width / 2) << "29" << std::endl;
```

Right justified table (default) :

Lastname	Firstname	Age
Daniel	Gray	25
Stanley	Woods	33
Jordan	Parker	45
Joe	Ball	21
Josh	Carr	27
Izaiah	Robinson	29

Left justified table :

Lastname	Firstname	Age
Daniel	Gray	25
Stanley	Woods	33
Jordan	Parker	45
Joe	Ball	21
Josh	Carr	27
Izaiah	Robinson	29

iostream

A screenshot of a terminal window on a dark-themed Mac OS X system. The window title bar is black with red, yellow, and green close/minimize/maximize buttons. The main pane is dark gray. In the top-left corner of the terminal area, there are three small colored dots (red, yellow, green). The terminal displays the following C++ code:

```
// Internal justified : sign is left justified , data is right justified
std::cout << std::endl;
std::cout << "Internal justified : " << std::endl;
std::cout << std::right;
std::cout << std::setw(10) << -123.45 << std::endl;
std::cout << std::internal;
std::cout << std::setw(10) << -123.45 << std::endl;
```

The output of the code is displayed in a separate window titled "Internal justified :". The window has a dark gray background and contains the following text:

Internal justified :
-123.45
- 123.45

iostream

```
//Set fill character
std::cout << std::left;
std::cout << std::setfill('*'); // Fill character

// First row (headers)
std::cout << std::setw(col_width) << "Lastname"
<< std::setw(col_width) << "Firstname"
<< std::setw(col_width / 2) << "Age"
<< std::endl;

// Data rows
std::cout << std::setw(col_width) << "Daniel"
<< std::setw(col_width) << "Gray"
<< std::setw(col_width / 2) << "25"
<< std::endl;

std::cout << std::setw(col_width) << "Stanley"
<< std::setw(col_width) << "Woods"
<< std::setw(col_width / 2) << "33"
<< std::endl;

std::cout << std::setw(col_width) << "Jordan"
<< std::setw(col_width) << "Parker"
<< std::setw(col_width / 2) << "45"
<< std::endl;
```

Table with fill characters :

Lastname	Firstname	Age
*****Daniel*****	*****Gray*****	*****25*****
*****Stanley*****	*****Woods*****	*****33*****
*****Jordan*****	*****Parker*****	*****45*****
*****Joe*****	*****Ball*****	*****21*****
*****Josh*****	*****Carr*****	*****27*****
*****Izaiah*****	*****Robinson*****	*****29*****

iostream

```
//boolalpha and noboolalpha : control bool output format : 1/0 or true/false
bool condition{ true };
bool other_condition{ false };

std::cout << "condition : " << condition << std::endl;
std::cout << "other_condition : " << other_condition << std::endl;

std::cout << std::endl;
std::cout << std::boolalpha; // 1/0
std::cout << "condition : " << condition << std::endl;
std::cout << "other_condition : " << other_condition << std::endl;

std::cout << std::endl;
std::cout << std::noboolalpha; // true/false
std::cout << "condition : " << condition << std::endl;
std::cout << "other_condition : " << other_condition << std::endl;
```

```
condition : 1
other_condition : 0
```

```
condition : true
other_condition : false
```

```
condition : 1
other_condition : 0
```

iostream

```
//showpos and noshowpos : show or hide the + sign for positive numbers
int pos_num{ 34 };
int neg_num{ -45 };

std::cout << "pos_num : " << pos_num << std::endl;
std::cout << "neg_num : " << neg_num << std::endl;

std::cout << std::endl;
std::cout << std::showpos;
std::cout << "pos_num : " << pos_num << std::endl;
std::cout << "neg_num : " << neg_num << std::endl;

std::cout << std::endl;
std::cout << std::noshowpos;
std::cout << "pos_num : " << pos_num << std::endl;
std::cout << "neg_num : " << neg_num << std::endl;
```

```
pos_num : 34
neg_num : -45

pos_num : +34
neg_num : -45

pos_num : 34
neg_num : -45
```

iostream



```
// uppercase and nouppercase
pos_int = 717171;

std::cout << "pos_int (nouppercase : default) : " << std::endl;
std::cout << "pos_int (dec) : " << std::dec << pos_int << std::endl;
std::cout << "pos_int (hex) : " << std::hex << pos_int << std::endl;
std::cout << "pos_int (oct) : " << std::oct << pos_int << std::endl;

std::cout << std::endl;
std::cout << "pos_int (uppercase) : " << std::endl;
std::cout << std::uppercase;
std::cout << "pos_int (dec) : " << std::dec << pos_int << std::endl;
std::cout << "pos_int (hex) : " << std::hex << pos_int << std::endl;
std::cout << "pos_int (oct) : " << std::oct << pos_int << std::endl;
```

```
pos_int (nouppercase : default) :
pos_int (dec) : 717171
pos_int (hex) : af173
pos_int (oct) : 2570563

pos_int (uppercase) :
pos_int (dec) : 717171
pos_int (hex) : AF173
pos_int (oct) : 2570563
```

iostream

```
// different number systems : std::dec, std::hex, std::oct
int pos_int{ 717171 };
int neg_int{ -47347 };
double double_var{ 498.32 };

std::cout << std::endl;
std::cout << "default base format : " << std::endl;
std::cout << "pos_int : " << pos_int << std::endl;
std::cout << "neg_int : " << neg_int << std::endl;
std::cout << "double_var : " << double_var << std::endl;

std::cout << std::endl;
std::cout << "pos_int in different bases : " << std::endl;
std::cout << "pos_int (dec) : " << std::dec << pos_int << std::endl;
std::cout << "pos_int (hex) : " << std::hex << pos_int << std::endl;
std::cout << "pos_int (oct) : " << std::oct << pos_int << std::endl;

    std::cout << std::endl;
std::cout << "neg_int in different bases : " << std::endl;
std::cout << "neg_int (dec) : " << std::dec << neg_int << std::endl;
std::cout << "neg_int (hex) : " << std::hex << neg_int << std::endl;
std::cout << "neg_int (oct) : " << std::oct << neg_int << std::endl;

std::cout << std::endl;
std::cout << "double_var in different bases : " << std::endl;
std::cout << "double_var (dec) : " << std::dec << double_var << std::endl;
std::cout << "double_var (hex) : " << std::hex << double_var << std::endl;
std::cout << "double_var (oct) : " << std::oct << double_var << std::endl;
```

```
default base format :
pos_int : 717171
neg_int : -47347
double_var : 498.32
```

```
pos_int in different bases :
pos_int (dec) : 717171
pos_int (hex) : af173
pos_int (oct) : 2570563
```

```
neg_int in different bases :
neg_int (dec) : -47347
neg_int (hex) : fffff470d
neg_int (oct) : 37777643415
```

```
double_var in different bases :
double_var (dec) : 498.32
double_var (hex) : 498.32
double_var (oct) : 498.32
```

```
pos_int (nouppercase : default) :
pos_int (dec) : 717171
pos_int (hex) : af173
pos_int (oct) : 2570563
```

```
pos_int (uppercase) :
pos_int (dec) : 717171
pos_int (hex) : AF173
pos_int (oct) : 2570563
```

iostream



```
// fixed and scientific : for floating point values
double a{ 3.1415926535897932384626433832795 };
double b{ 2006.0 };
double c{ 1.34e-10 };

std::cout << std::endl;
std::cout << "double values (default : use scientific where necessary) : " << std::endl;
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;

std::cout << std::endl;
std::cout << "double values (fixed) : " << std::endl;
std::cout << std::fixed;
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;

std::cout << std::endl;
std::cout << "double values (scientific) : " << std::endl;
std::cout << std::scientific;
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;
```

double values (default : use scientific where necessary) :
a : 3.14159
b : 2006
c : 1.34E-10

double values (fixed) :
a : 3.141593
b : 2006.000000
c : 0.000000

double values (scientific) :
a : 3.141593E+00
b : 2.006000E+03
c : 1.340000E-10

double values (back to defaults) :
a : 3.14159
b : 2006
c : 1.34E-10

iostream

```
// fixed and scientific : for floating point values
double a{ 3.1415926535897932384626433832795 };
double b{ 2006.0 };
double c{ 1.34e-10 };

std::cout << std::endl;
std::cout << "double values (default: use scientific where necessary) : " << std::endl;
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;

std::cout << std::endl;
std::cout << "double values (fixed): " << std::endl;
std::cout << std::fixed;
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;

std::cout << std::endl;
std::cout << "double values (scientific): " << std::endl;
std::cout << std::scientific;
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;

std::cout << "double values (back to defaults): " << std::endl;
std::cout.unsetf(std::ios::scientific | std::ios::fixed); // Hack
std::cout << "a : " << a << std::endl;
std::cout << "b : " << b << std::endl;
std::cout << "c : " << c << std::endl;
```

```
double values (default : use scientific where necessary) :
a : 3.14159
b : 2006
c : 1.34E-10

double values (fixed) :
a : 3.141593
b : 2006.000000
c : 0.000000

double values (scientific) :
a : 3.141593E+00
b : 2.006000E+03
c : 1.340000E-10

double values (back to defaults) :
a : 3.14159
b : 2006
c : 1.34E-10
```

iostream

```
// setprecision() : the number of digits printed out for a floating point. Default is 6
a = 3.1415926535897932384626433832795;

std::cout << std::endl;
std::cout << "a (default precision(6)) : " << a << std::endl;
std::cout << std::setprecision(10);
std::cout << "a (precision(10)) : " << a << std::endl;
std::cout << std::setprecision(20);
std::cout << "a (precision(20)) : " << a << std::endl;
```

```
a (default precision(6)) : 3.14159
a (precision(10)) : 3.141592654
a (precision(20)) : 3.141592653589793116
```

iostream

```
// showpoint and noshowpoint : show trailing zeros if necessary
// Force output of the decimal point
double d{ 34.1 };
double e{ 101.99 };
double f{ 12.0 };
int g{ 45 };

std::cout << std::endl;
std::cout << "noshowpoint (default) : " << std::endl;
std::cout << "d : " << d << std::endl;
std::cout << "e : " << e << std::endl;
std::cout << "f : " << f << std::endl;// 12
std::cout << "g : " << g << std::endl;

std::cout << std::endl;
std::cout << "showpoint: " << std::endl;
std::cout << std::showpoint;
std::cout << "d : " << d << std::endl;
std::cout << "e : " << e << std::endl;
std::cout << "f : " << f << std::endl;// 12.0
std::cout << "g : " << g << std::endl;
```

```
noshowpoint (default) :
d : 34.1
e : 101.99
f : 12
g : 55

showpoint:
d : 34.1000
e : 101.990
f : 12.0000
g : 55
```