C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
Containers:
Input/Output:
Multi-threading:
Other:
<algorithm>
<bitset>
<chrono> C++11
<codecvt> C++11
<complex>
<exception>
<functional>
<initializer_list> C++11
<iterator>
<limits>
<locale>
<memory>
<new>
<numeric>
<random> C++11
<ratio> C++11
<regex> C++11
<stdexcept>
<string>
<system_error> C++11
<tuple> C++11
<typeindex> C++11
<typeinfo>
<type_traits> C++11
<utility>
<valarray>

<bitset>
bitset

bitset
bitset::bitset
bitset operators
member classes:
bitset::reference
member functions:
bitset::all C++11
bitset::any
bitset::count
bitset::flip
bitset::none
bitset::operator[]
bitset::reset
bitset::set
bitset::size
bitset::test
bitset::to_string
bitset::to_ullong C++11
bitset::to_ulong
non-member specializations:
hash<bitset> C++11

public member function

std::bitset::bitset

<bitset>

C++98 C++11

default (1)	`bitset();`
integer value (2)	`bitset (unsigned long val);`
string (3)	`template<class charT, class traits, class Alloc> explicit bitset (const basic_string<charT,traits,Alloc>& str, typename basic_string<charT,traits,Alloc>::size_type pos = 0, typename basic_string<charT,traits,Alloc>::size_type n = basic_string<charT,traits,Alloc>::npos);`

Construct bitset

Constructs a `bitset` container object:

(1) default constructor
The object is initialized with zeros.

(2) initialization from integer value
Initializes the object with the bit values of *val*:

(3) initialization from string or (4) C-string
Uses the sequence of *zeros* and/or *ones* in *str* to initialize the first *n* bit positions of the constructed `bitset` object.

Note that `bitset` objects have a *fixed size* (determined by their class template argument) no matter the constructor used: Those bit positions not explicitly set by the constructor are initialized with a value of zero.

Parameters

`val`
Integral value whose bits are copied to the bitset positions.
- If the value representation of *val* is greater than the *bitset size*, only the least significant bits of *val* are taken into consideration.
- If the value representation of *val* is less than the *bitset size*, the remaining bit positions are initialized to zero.

`str`
C++98 C++11
A `basic_string` whose contents are used to initialize the `bitset`:
The constructor parses the string reading at most *n* characters beginning at *pos*, interpreting the character values `'0'` and `'1'` as zero and one, respectively.
Note that the least significant bit is represented by the last character read (not the first); Thus, the first bit position is read from the right-most character, and the following bits use the characters preceding this, from right to left.
If this sequence is shorter than the *bitset size*, the remaining bit positions are initialized to zero.

`pos`
First character in the `basic_string` to be read and interpreted.
If this is greater than the *length* of *str*, an `out_of_range` exception is thrown.

`n`
Number of characters to read. Any value greater than the *bitset size* (including `npos`) is equivalent to specifying exactly the *bitset size*.

`zero, one`
Character values to represent *zero* and *one*.

Example

```cpp
// constructing bitsets
#include <iostream>       // std::cout
#include <string>         // std::string
#include <bitset>         // std::bitset

int main ()
{
  std::bitset<16> foo;
  std::bitset<16> bar (0xfa2);
  std::bitset<16> baz (std::string("0101111001"));

  std::cout << "foo: " << foo << '\n';
  std::cout << "bar: " << bar << '\n';
  std::cout << "baz: " << baz << '\n';

  return 0;
}
```

Output:

```
foo: 0000000000000000
bar: 0000111110100010
baz: 0000000101111001
```

Data races

Constructors *(3)* and *(4)* access the characters in *str*.

Exception safety

Neither the *default constructor (1)* nor the *constructor from integer value (2)* throw exceptions.
The other constructors cause no side effects in case an exception is thrown (strong guarantee).
Throws `out_of_range` if `pos > str.size()`.

See also

bitset::set	Set bits (public member function)
bitset::reset	Reset bits (public member function)
bitset::operator[]	Access bit (public member function)
bitset operators	Bitset operators (function)