**std::bitset**

A bitset stores bits (elements with only two possible values: 0 or 1, true or false,).

**Here index count from backward**

*If, b=(string(“10011”));*

*Then b[0]=1 b[1]=1 b[2]=0 b[3]-=0 b[4]=1*

**The** [**size**](http://www.cplusplus.com/bitset::size) **of a bitset is fixed at compile-time (determined by its template parameter).**

**Constructor:**

*std::bitset<16> foo;//all values are set to 0*

*std::bitset<16> bar (0xfa2); //integer value*

*std::bitset<16> baz (std::string("0101111001")) //string value*

**Operators:**

We can use all bitwise operator with it

**Functions(public class):**

**Bit access** [**operator[]**](http://127.0.0.1/reference/bitset/bitset/operator%5B%5D/) we can access or set any value. example b[1]=c[2]

Access bit

[**count**](http://www.cplusplus.com/reference/bitset/bitset/count/)**():** Count bits set

[**size**](http://www.cplusplus.com/reference/bitset/bitset/size/)(): Return size

[**test**](http://www.cplusplus.com/reference/bitset/bitset/test/)**(position)**: Return bit value if

*if, b=(string(“10011”)); Then b.test(1) will return*

[**any**](http://www.cplusplus.com/reference/bitset/bitset/any/)**():** Test if any bit is set

[**none**](http://www.cplusplus.com/reference/bitset/bitset/none/)**():** Test if no bit is set

**all():** Test if all bits are set

[**set**](http://www.cplusplus.com/reference/bitset/bitset/set/)(): Set all bits

*std::bitset<4> foo;*

*std::cout << foo.set() << '\n';* // 1111

*std::cout << foo.set(2,0) << '\n';* // 1011

*std::cout << foo.set(2) << '\n';* // 1111

[**reset**](http://www.cplusplus.com/reference/bitset/bitset/reset/)(): Reset all bits

*std::bitset<4> foo (std::string("1011"));*

*std::cout << foo.reset(1) << '\n';* // 1001

*std::cout << foo.reset() << '\n';* // 0000

[**flip**](http://www.cplusplus.com/reference/bitset/bitset/flip/)(): Flip bits

*std::bitset<4> foo (std::string("0001"));*

*std::cout << foo.flip(2) << '\n';* // 0101

*std::cout << foo.flip() << '\n';* // 1010

[**to\_string**](http://www.cplusplus.com/reference/bitset/bitset/to_string/)(): Convert to string

[**to\_ulong**](http://www.cplusplus.com/reference/bitset/bitset/to_ulong/)(): Convert to unsigned long integer

[**to\_ullong()**](http://www.cplusplus.com/reference/bitset/bitset/to_ullong/) : Convert to unsigned long long

Top of Form

Bottom of Form

Top of Form

Bottom of Form