

```

/* The following code example is taken from the book
 * "The C++ Standard Library – A Tutorial and Reference"
 * by Nicolai M. Josuttis, Addison-Wesley, 1999
 *
 * (C) Copyright Nicolai M. Josuttis 1999.
 * Permission to copy, use, modify, sell and distribute this software
 * is granted provided this copyright notice appears in all copies.
 * This software is provided "as is" without express or implied
 * warranty, and with no claim as to its suitability for any purpose.
 */
#include <cstddef>

template<class T, std::size_t thesize>
class carray {
private:
    T v[thesize];    // fixed-size array of elements of type T

public:
    // type definitions
    typedef T          value_type;
    typedef T*         iterator;
    typedef const T*   const_iterator;
    typedef T&         reference;
    typedef const T&   const_reference;
    typedef std::size_t size_type;
    typedef std::ptrdiff_t difference_type;

    // iterator support
    iterator begin() { return v; }
    const_iterator begin() const { return v; }
    iterator end() { return v+thesize; }
    const_iterator end() const { return v+thesize; }

    // direct element access
    reference operator[](std::size_t i) { return v[i]; }
    const_reference operator[](std::size_t i) const { return v[i]; }

    // size is constant
    size_type size() const { return thesize; }
    size_type max_size() const { return thesize; }

    // conversion to ordinary array
    T* as_array() { return v; }
};

```