

```

/* The following code example is taken from the book
 * "The C++ Standard Library – A Tutorial and Reference, 2nd Edition"
 * by Nicolai M. Josuttis, Addison-Wesley, 2012
 *
 * (C) Copyright Nicolai M. Josuttis 2012.
 * 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.
 */
/* *****
 * Queue.hpp
 * – safer and more convenient queue class
 * *****/
#ifndef QUEUE_HPP
#define QUEUE_HPP

#include <deque>
#include <exception>

template <typename T>
class Queue {
protected:
    std::deque<T> c;          // container for the elements

public:
    // exception class for pop() and front() with empty queue
    class ReadEmptyQueue : public std::exception {
    public:
        virtual const char* what() const throw() {
            return "read empty queue";
        }
    };

    // number of elements
    typename std::deque<T>::size_type size() const {
        return c.size();
    }

    // is queue empty?
    bool empty() const {
        return c.empty();
    }

    // insert element into the queue
    void push (const T& elem) {
        c.push_back(elem);
    }

    // remove next element from the queue and return its value
    T pop () {
        if (c.empty()) {
            throw ReadEmptyQueue();
        }
        T elem(c.front());
        c.pop_front();
    }
};

```

```
        return elem;
    }

    // return value of next element
    T& front () {
        if (c.empty()) {
            throw ReadEmptyQueue();
        }
        return c.front();
    }
};

#endif /* QUEUE_HPP */
```