```cpp
/* 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.
 */
/* **********************************************************
 *  Stack.hpp
 *   - safer and more convenient stack class
 * *********************************************************/
#ifndef STACK_HPP
#define STACK_HPP

#include <deque>
#include <exception>

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

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

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

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

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

    // pop element out of the stack and return its value
    T pop () {
        if (c.empty()) {
            throw ReadEmptyStack();
        }
        T elem(c.back());
        c.pop_back();
```

```cpp
            return elem;
        }

        // return value of next element
        T& top () {
            if (c.empty()) {
                throw ReadEmptyStack();
            }
            return c.back();
        }
};

#endif /* STACK_HPP */
```