```
/* 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.
 */
#include <thread>
#include <future>
#include <iostream>
#include <string>
#include <exception>
#include <stdexcept>
#include <functional>
#include <utility>

void doSomething (std::promise<std::string>& p)
{
    try {
        // read character and throw exceptiopn if 'x'
        std::cout << "read char ('x' for exception): ";
        char c = std::cin.get();
        if (c == 'x') {
            throw std::runtime_error(std::string("char ")+c+" read");
        }
        //...
        std::string s = std::string("char ") + c + " processed";
        p.set_value(std::move(s));    // store result
    }
    catch (...) {
        p.set_exception(std::current_exception());  // store exception
    }
}

int main()
{
    try {
        // start thread using a promise to store the outcome
        std::promise<std::string> p;
        std::thread t(doSomething,std::ref(p));
        t.detach();
        //...

        // create a future to process the outcome
        std::future<std::string> f(p.get_future());

        // process the outcome
        std::cout << "result: " << f.get() << std::endl;
    }
    catch (const std::exception& e) {
        std::cerr << "EXCEPTION: " << e.what() << std::endl;
    }
    catch (...) {
        std::cerr << "EXCEPTION " << std::endl;
    }
}
```