```
/* 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 <string>
#include <iostream>
#include <locale>
#include <exception>
#include <cstdlib>
using namespace std;

// output operator for pos_format() and neg_format():
ostream& operator<< (ostream& strm, moneypunct<char>::pattern p)
{
    for (int i=0; i<4; ++i) {
        auto f = p.field[i];
        strm << (f==money_base::none ?   "none" :
                 f==money_base::space ?  "space" :
                 f==money_base::symbol ? "symbol" :
                 f==money_base::sign ?   "sign" :
                 f==money_base::value ?  "value" :
                 "???") << " ";
    }
    return strm;
}

template <bool intl>
void printMoneyPunct (const string& localeName)
{
    locale loc(localeName);
    const moneypunct<char,intl>& mp
            = use_facet<moneypunct<char,intl>>(loc);
    cout << "moneypunct in locale \"" << loc.name() << "\":" << endl;
    cout << " decimal_point: " << (mp.decimal_point()!='\0' ?
                                   mp.decimal_point() : ' ') << endl;
    cout << " thousands_sep: " << (mp.thousands_sep()!='\0' ?
                                   mp.thousands_sep() : ' ') << endl;
    cout << " grouping:      ";
    for (int i=0; i<mp.grouping().size(); ++i) {
        cout << static_cast<int>(mp.grouping()[i]) << ' ';
    }
    cout << endl;
    cout << " curr_symbol:   " << mp.curr_symbol() << endl;
    cout << " positive_sign: " << mp.positive_sign() << endl;
    cout << " negative_sign: " << mp.negative_sign() << endl;
    cout << " frac_digits:   " << mp.frac_digits() << endl;
    cout << " pos_format:    " << mp.pos_format() << endl;
    cout << " neg_format:    " << mp.neg_format() << endl;
}
```

```
int main ()
{
    try {
        printMoneyPunct<false>("C");
        cout << endl;
        printMoneyPunct<false>("german");
        cout << endl;
        printMoneyPunct<true>("german");
    }
    catch (const std::exception& e) {
        cerr << "Exception: " << e.what() << endl;
        return EXIT_FAILURE;
    }
}
```