```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.
 */
#include <iostream>
#include <iomanip>
#include <utility>
#include <iterator>
#include <typeinfo>

// generic output for pairs (map elements)
template <typename T1, typename T2>
std::ostream& operator << (std::ostream& strm, const std::pair<T1,T2>& p)
{
    return strm << "[" << p.first << "," << p.second << "]";
}

template <typename T>
void printHashTableState (const T& cont)
{
    // basic layout data:
    std::cout << "size:            " << cont.size() << "\n";
    std::cout << "buckets:         " << cont.bucket_count() << "\n";
    std::cout << "load factor:     " << cont.load_factor() << "\n";
    std::cout << "max load factor: " << cont.max_load_factor() << "\n";

    // iterator category:
    if (typeid(typename std::iterator_traits
                          <typename T::iterator>::iterator_category)
        == typeid(std::bidirectional_iterator_tag)) {
          std::cout << "chaining style:  doubly-linked" << "\n";
    }
    else {
          std::cout << "chaining style:  singly-linked" << "\n";
    }

    // elements per bucket:
    std::cout << "data: " << "\n";
    for (auto idx=0; idx != cont.bucket_count(); ++idx) {
            std::cout << " b[" << std::setw(2) << idx << "]: ";
            for (auto pos=cont.begin(idx); pos != cont.end(idx); ++pos) {
                std::cout << *pos << " ";
            }
            std::cout << "\n";
    }
    std::cout << std::endl;
}
```