```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 <map>
#include <string>
#include <algorithm>
#include <cctype>
using namespace std;

// function object to compare strings
// - allows you to set the comparison criterion at runtime
// - allows you to compare case insensitive
class RuntimeStringCmp {
  public:
    // constants for the comparison criterion
    enum cmp_mode {normal, nocase};
  private:
    // actual comparison mode
    const cmp_mode mode;

    // auxiliary function to compare case insensitive
    static bool nocase_compare (char c1, char c2) {
        return toupper(c1) < toupper(c2);
    }
  public:
    // constructor: initializes the comparison criterion
    RuntimeStringCmp (cmp_mode m=normal) : mode(m) {
    }

    // the comparison
    bool operator() (const string& s1, const string& s2) const {
        if (mode == normal) {
            return s1<s2;
        }
        else {
            return lexicographical_compare (s1.begin(), s1.end(),
                                            s2.begin(), s2.end(),
                                            nocase_compare);
        }
    }
};

// container type:
// - map with
//       - string keys
//       - string values
//       - the special comparison object type
```

```
typedef map<string,string,RuntimeStringCmp> StringStringMap;

// function that fills and prints such containers
void fillAndPrint(StringStringMap& coll);

int main()
{
    // create a container with the default comparison criterion
    StringStringMap coll1;
    fillAndPrint(coll1);

    // create an object for case-insensitive comparisons
    RuntimeStringCmp ignorecase(RuntimeStringCmp::nocase);

    // create a container with the case-insensitive comparisons criterion
    StringStringMap coll2(ignorecase);
    fillAndPrint(coll2);
}

void fillAndPrint(StringStringMap& coll)
{
    // insert elements in random order
    coll["Deutschland"] = "Germany";
    coll["deutsch"] = "German";
    coll["Haken"] = "snag";
    coll["arbeiten"] = "work";
    coll["Hund"] = "dog";
    coll["gehen"] = "go";
    coll["Unternehmen"] = "enterprise";
    coll["unternehmen"] = "undertake";
    coll["gehen"] = "walk";
    coll["Bestatter"] = "undertaker";

    // print elements
    cout.setf(ios::left, ios::adjustfield);
    for (const auto& elem : coll) {
        cout << setw(15) << elem.first << " "
             << elem.second << endl;
    }
    cout << endl;
}
```