```
/* The following code example is taken from the book
 * "The C++ Standard Library - A Tutorial and Reference"
 * by Nicolai M. Josuttis, Addison-Wesley, 1999
 *
 * (C) Copyright Nicolai M. Josuttis 1999.
 * 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 <set>
#include "print.hpp"
using namespace std;

// type for sorting criterion
template <class T>
class RuntimeCmp {
  public:
    enum cmp_mode {normal, reverse};
  private:
    cmp_mode mode;
  public:
    // constructor for sorting criterion
    // - default criterion uses value normal
    RuntimeCmp (cmp_mode m=normal) : mode(m) {
    }
    // comparison of elements
    bool operator() (const T& t1, const T& t2) const {
        return mode == normal ? t1 < t2 : t2 < t1;
    }
    // comparison of sorting criteria
    bool operator== (const RuntimeCmp& rc) {
        return mode == rc.mode;
    }
};

// type of a set that uses this sorting criterion
typedef set<int,RuntimeCmp<int> > IntSet;

// forward declaration
void fill (IntSet& set);

int main()
{
    // create, fill, and print set with normal element order
    // - uses default sorting criterion
    IntSet coll1;
    fill(coll1);
    PRINT_ELEMENTS (coll1, "coll1: ");

    // create sorting criterion with reverse element order
    RuntimeCmp<int> reverse_order(RuntimeCmp<int>::reverse);

    // create, fill, and print set with reverse element order
    IntSet coll2(reverse_order);
```

```
    fill(coll2);
    PRINT_ELEMENTS (coll2, "coll2: ");

    // assign elements AND sorting criterion
    coll1 = coll2;
    coll1.insert(3);
    PRINT_ELEMENTS (coll1, "coll1: ");

    // just to make sure...
    if (coll1.value_comp() == coll2.value_comp()) {
        cout << "coll1 and coll2 have same sorting criterion"
             << endl;
    }
    else {
        cout << "coll1 and coll2 have different sorting criterion"
             << endl;
    }
}

void fill (IntSet& set)
{
    // fill insert elements in random order
    set.insert(4);
    set.insert(7);
    set.insert(5);
    set.insert(1);
    set.insert(6);
    set.insert(2);
    set.insert(5);
}
```