```
/* 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 <string>
#include <set>
#include <deque>
#include <algorithm>
#include <memory>

class Item {
  private:
    std::string name;
    float  price;
  public:
    Item (const std::string& n, float p = 0) : name(n), price(p) {
    }
    std::string getName () const {
        return name;
    }
    void setName (const std::string& n) {
        name = n;
    }
    float getPrice () const {
        return price;
    }
    float setPrice (float p) {
        price = p;
    }
};

template <typename Coll>
void printItems (const std::string& msg, const Coll& coll)
{
    std::cout << msg << std::endl;
    for (const auto& elem : coll) {
        std::cout << ' ' << elem->getName() << ": "
                  << elem->getPrice() << std::endl;
    }
}

int main()
{
    using namespace std;

    // two different collections sharing Items
    typedef shared_ptr<Item> ItemPtr;
    set<ItemPtr> allItems;
    deque<ItemPtr> bestsellers;
    // insert objects into the collections
    // - bestsellers are in both collections
    bestsellers = { ItemPtr(new Item("Kong Yize",20.10)),
                    ItemPtr(new Item("A Midsummer Night's Dream",14.99)),
                    ItemPtr(new Item("The Maltese Falcon",9.88)) };
    allItems = { ItemPtr(new Item("Water",0.44)),
                 ItemPtr(new Item("Pizza",2.22)) };
    allItems.insert(bestsellers.begin(),bestsellers.end());

    // print contents of both collections
    printItems ("bestsellers:", bestsellers);
    printItems ("all:", allItems);
    cout << endl;

    // double price of bestsellers
    for_each (bestsellers.begin(), bestsellers.end(),
              [] (shared_ptr<Item>& elem) {
                  elem->setPrice(elem->getPrice() * 2);
              });

    // replace second bestseller by first item with name "Pizza"
    bestsellers[1] = *(find_if(allItems.begin(),allItems.end(),
                               [] (shared_ptr<Item> elem) {
                                   return elem->getName() == "Pizza";
                               }));

    // set price of first bestseller
    bestsellers[0]->setPrice(44.77);

    // print contents of both collections
    printItems ("bestsellers:", bestsellers);
    printItems ("all:", allItems);
}
```