Sorting an Array of Object

Given a class declaration for book below:

```
1.  class book
2.  {
3.    private:
4.    // data members should be private
5.      float price;
6.      int year;
7.      char author[20], title[25];
8.    public:
9.      void getData();
10.     void print( );
11.     char * getTitle() {return title;}
12.     float getPrice() { return price;}
13. };  // end book declaration
```

We can have an array of book as declared in main() below and sort the array using any sorting technique. In this example, the array is sorted using Bubble sort.

```
1.  main ( )
2.  {   book myBook[size];
3.      for (int j=0;j<size;j++)
4.          myBook[j].getData();
5.      for (int j=0;j<size;j++)
6.          myBook[j].print();
7.      // sort the array using bubble sort
8.      BubbleSort(myBook,size) ;
9.      cout << "\nThe list after sort based on the price -> \n";
10.     for (int j=0;j<size;j++)
11.         myBook[j].print();
12.     return 0;
13. }// end main()
```

Bubble sort function that sort the array based on the book's price..

```
1.  void BubbleSort(book data[],int listSize)
2.  { int x ;
3.    book tempValue;
4.  for ( int pass =1;pass < listSize; pass++ )
5.  { for ( int x = 0; x < listSize - pass; x++)
6.     { //compare adjacent list based on the book's price
7.       if (data[x].getPrice()> data[x+1].getPrice())
8.       // swap if the data is not in the right order
9.       { tempValue = data[x];
10.        data[x] = data[x+1];
11.        data[x+1] = tempValue;
12.      }// end if
13.    } // end for loop (internal)
14. } // end for loop (external)
15. } // end BubbleSort() function
```

You can also sort the array of books based on the book's title as shown in the example below:

```
          { //compare adjacent list based on the book's title
      if ((strcmp(data[x].getTitle(),data[x+1].getTitle()))>0)
        // swap if the data is not in the right order
       { tempValue = data[x];
         data[x] = data[x+1];
         data[x+1] = tempValue;
       }// end if
```

Searching from an array of Book.

The following example search a book based on the book's title. Assumed that the list is sorted based on book's title and searching algorithm being used is sequential search on sorted data.)

```
1.  int SortedSeqSearch (book array[ ], int array_size, char search_key[])
2.  { int p;
3.    int index = -1; //-1 means record not found yet
4.    for ( p = 0; p < array_size; p++ )
5.    {
6.      if ((strcmp(search_key,array [p].getTitle())) < 0 )
7.          break; // loop repetition terminated when the value of
8.        // search key is smaller than the current array element
9.      else if ((strcmp(search_key, array [p].getTitle())) == 0)
10.     { index = p; // found at current array index
11.       break; // stop searching
12.     } // end else-if
13.   }//end for
14. return index;
15. } //end SortedSeqSearch() function
16.
```

```
1.  main ( )
2.  {   book myBook[size];
3.      for (int j=0;j<size;j++)
4.          myBook[j].getData();
5.
6.      // sort the array using bubble sort
7.      BubbleSort(myBook,size) ;
8.      cout << "\nPlease insert the book's title to be searched…";
9.      cin.getline(search_key,25);
10.     // search the book
11.     if (SortedSeqSearch (myBook, size, search_key)== -1)
12.        cout << "\n Sorry the book is not found.";
13.     else
14.        cout << "\n Yes, we have the book in our stock.";
15.
16.     return 0;
17. }// end main()
```