

**CALIFORNIA STATE UNIVERSITY, SACRAMENTO**

College of Business Administration

MIS 211 - Information Systems II

C++ Answer to Homework 3

**Main Function**

```

#include <fstream.h>
#include <iostream.h>
#include <stdlib.h>
#include <iomanip.h>
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <direct.h>
//
class videos          - Class declaration
{
    protected:
        double      price[100], stock_num[100];  - Arrays of the class
        int         i, elements;

    public:
        void AssignVal(double, double, int);     - Methods (functions)
        double PricePeek(short);
        double StockNumPeek(short);
};

void videos::AssignVal(double a, double b, int i)  - Method to load price and
                                                    stock number arrays
{
    stock_num[i] = a;
    price[i] = b;
    elements = i + 1;
}

//----- Price Peek -----
double videos::PricePeek(short i)                 - Method to retrieve prices
{
    return price[i];
}

//----- Stock Num Peek -----
double videos::StockNumPeek(short i)              - Method to retrieve stock
                                                    numbers
{
    return stock_num[i];
}

```



```

class seq_search : public videos           - Subclass declaration
{
}
;

class bin_search : public videos          - Subclass declaration
{
}
;

//-----
int main()
{
    char  vid_title[41], vid_title_array[100][41];
    double      stk_num = 0, price = 0, s_num = 0;
    short count = 0, i = 0;

    seq_search seq_object;                 - Instantiate seq_object in seq_search subclass
    bin_search bin_object;                 - Instantiate bin_object in bin_search subclass

    ifstream infile;
    infile.open("videos.dat");

// Records from the videos file are sequentially loaded into the arrays

    infile >> stk_num;                      - Priming reads
    infile.getline(vid_title,41);
    infile >> price;

    while (!infile.eof())                  - Begin loop to load arrays
    {
        strcpy(vid_title_array[count],vid_title); - Load video titles into array
        bin_object.AssignVal(stk_num,price,count); - Load prices into array
        seq_object.AssignVal(stk_num,price,count); - Load stock numbers into
        count += 1;                          array

        infile >> stk_num;                    - Second reads to retrieve next
        infile.getline(vid_title,41);         records
        infile >> price;

    }                                         - End loop

    cout << count << "\n";

    ifstream numfile;
    ofstream output;

    numfile.open("stocknum.dat");
    output.open("report.txt");

```



```

output      << setw(10) << " "
           << "Stock"
           << "\n";

output << setw(10) << " "
           << "Number"
           << setw(13) << "Video Title"
           << setw(20) << " "
           << setw(18) << "Price"
           << "\n\n";

// Stock numbers, videos titles and prices are sequentially retrieved
// from their respective arrays

for (i = 0; i < count; ++i)           - Begin loop to retrieve
{
    output << setw(5) << i + 1
           << setw(10)
           << bin_object.StockNumPeek(i)  - Retrieve stock number
           << setw(2) << " "
           << strcpy(vid_title,vid_title_array[i])  - Retrieve video title
           << setw(10)
           << bin_object.PricePeek(i)      - Retrieve price
           << "\n";

}                                       - End loop
infile.close();
numfile.close();
output.close();
return 0;
}

```



**Output**

	Stock Number	Video Title	Price
1	25	Kronos	19.95
2	100	Little Mermaid, The	26.95
3	410	Cinderella	26.95
4	448	X-Files: Fight the Future, The	22.95
5	1001	Hello, Dolly!	19.95
6	1004	King and I, The	19.95
7	1015	Bimbo Movie Bash	14.95
8	1148	Quest for Fire	29.95
9	4721	Patton	19.95
10	4806	Fully Monty, The	14.95
11	5493	Planet of the Apes Collection	49.95
12	6026	Back to Bataan	14.95
13	6703	Toy Story	26.95
14	6809	Romeo and Juliet	14.95
15	7021	Hondo	19.95
16	7126	Scrooge	14.95
17	8590	Longest Day, The	29.95
18	8692	Butch Cassidy and the Sundance Kid	9.99
19	10484	Con Air	19.95
20	12865	Flubber	22.95
21	13176	Interview with a Vampire	14.95
22	13553	Scream 2	22.95
23	13559	Good Will Hunting	22.95
24	15029	Camelot	24.95
25	15296	World Without End	14.95
26	15371	Charlie Brown Thanksgiving, A	12.95
27	15390	Star Trek-The Movies: Set	99.95
28	16291	Lethal Weapon 3	14.95
29	16500	Batman and Robin	14.95
30	16714	Just Cause	9.99
31	21693	Mask of Zorro, The	24.95
32	25493	Sleepless in Seattle	14.95
33	34461	Last Seduction, The	14.95
34	35109	Damn Yankees	19.95
35	35199	Thorn Birds, The	99.95
36	35465	National Lampoon's Vacation: Set	39.95
37	37141	Blue Hawaii	14.95
38	66009	Sting, The	14.95
39	80130	Winds of War	139.95
40	83655	Amistad	19.95
41	109602	It's a Wonderful Life	14.95
42	111209	Calamity Jane	19.95
43	201490	The Russians Are Coming! ...	19.95
44	206138	Red River	14.95
45	207132	Tomorrow Never Dies	39.95
46	302312	Jazz Singer, The	19.95



Spring 2002

Dr. Ching

47	500058	Captains Courageous	19.95
48	505204	Wizard of Oz, The	19.95
49	700008	Dirty Dozen	14.95
50	906832	Red Corner	19.95



*Recyclable paper. Please recycle.  
Reduce, Reuse, Recycle*