

Q: A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message "Required copies not in stock" is displayed. Design a system using a class called Books with suitable member functions and Constructors. Implement C++ program for the system.

Answer:

Source Code:

```
#include<iostream>
#include<string>
#include<iomanip>
#define M 5
using namespace std;

class Books{
private:
    string author;
    string title;
    float price;
    string publisher;
    int stock_position;
    void friend add_to_array(Books &,Books,int);
    void friend bill(Books arr[],int);
    void friend search_by_author(Books arr[],string);
```

```
void friend search_by_title(Books arr[],string);
void friend show_all_books(Books arr[]);

public:
    Books(){}
    Books(string a, string t, float pr, string pub, int st){
        author = a;
        title=t;
        price=pr;
        publisher=pub;
        stock_position = st;
    }
    bool match(string,string);
    void details();
    float query(int);
    void update_price(float);
};
```

```
bool Books::match(string t, string auth){
    if(title==t && auth==author) return true;
    return false;
}

void Books::details(){
    cout<<"Title: "<<title<<endl<<"Author: "<<author<<endl;
    cout<<"Publisher: "<<publisher<<endl<<"Price: "<<price<<endl;
}
```

```
float Books::query(int q){
    if(q<=stock_position){
        stock_position-=q;
        return q*price;
    }
}
```

```

return 0.0;
}

void Books::update_price( float p){
    price=p;
}

void add_to_array(Books &a,Books b,int q){
    a=b;
    a.stock_position = q;
}

void bill(Books arr[],int n){
    float total=0;
    for(int i=0;i<n;i++){
        total+=arr[i].price * arr[i].stock_position;
        cout<<"Item: "<<i+1<<endl;
        cout<<"Title: "<<arr[i].title<<endl;
        cout<<"Author: "<<arr[i].author<<endl;
        cout<<"Price: "<<arr[i].price<<endl;
        cout<<"Quantity: "<<arr[i].stock_position<<endl;
    }
    cout<<"-----\nTotal: "<<total<<endl;
}

void show_all_books(Books arr[]){
    cout<<"_____ \n";
    cout<<"| Title | Author | Publisher | Price | Stock Position | \n";
    cout<<"_____ \n";
    for(int i=0;i<M;i++){
        cout<<left<<" | "<<setw(18)<<arr[i].title<<" | "<<setw(17)<<arr[i].author<<" | "<<setw(18)<<arr[i].publisher<<" | "<<setw(10)
        )<<arr[i].price<<" | "<<setw(14)<<arr[i].stock_position<<" | \n";
    }
}

```

```
cout<<"_____ \n";  
}  
}
```

```
void search_by_author(Books arr[],string a){  
    cout<<"_____ \n";  
    cout<<"| Title | Publisher | Price | Stock Position|\n";  
    cout<<"_____ \n";  
    for(int i=0;i<M;i++){  
        if(arr[i].author==a){  
  
            cout<<left<<"|"<<setw(18)<<arr[i].title<<"|"<<setw(18)<<arr[i].publisher<<"|"<<setw(10)<<arr[i].price<<"|"<<setw(14)  
<<arr[i].stock_position<<"|\n";  
            cout<<"_____ \n";  
        }  
    }  
}
```

```
void search_by_title(Books arr[],string a){  
    cout<<"_____ \n";  
    cout<<"| Author | Publisher | Price | Stock Position|\n";  
    cout<<"_____ \n";  
    for(int i=0;i<M;i++){  
        if(arr[i].title==a){  
  
            cout<<left<<"|"<<setw(18)<<arr[i].title<<"|"<<setw(18)<<arr[i].publisher<<"|"<<setw(10)<<arr[i].price<<"|"<<setw(14)  
<<arr[i].stock_position<<"|\n";  
            cout<<"_____ \n";  
        }  
    }  
}
```

```

int main(){

    Books arr[M],bought[M];
    string author,title,publisher;
    float price;
    int stock,n=0;
    bool flag;

    cout<<"Eneter Details of books in stock.\n";
    for(int i=0;i<M;i++){
        cout<<"Details of book "<<i+1<<endl;
        cout<<"Enter Title\n";
        cin.clear();
        cin.sync();
        getline(cin,title);
        cout<<"Enter Author's name\n";
        getline(cin,author);
        cout<<"Enter Publisher\n";
        getline(cin,publisher);
        cout<<"Enter Price\n";
        cin>>price;
        cout<<"Enter Stock Position\n";
        cin>>stock;
        arr[i]=Books(author,title,price,publisher,stock);
    }

    while(1){
        int choice=0;

        cout<<"Enter 1 to buy Book\nEnter 2 to see all book details\nEnter 3 to update price of a book\nEnter 4 to search
all books by an author\nEnter 5 to search books by Title\nEnter 0 to exit.\n";

        cin>>choice;
        switch(choice){

            case 1:
                cout<<"Enter the Title and Author of the book\n";
                cin.clear();

```

```

cin.sync();

getline(cin,title);
getline(cin,author);
for(int i=0;i<M;i++){
    flag = arr[i].match(title,author);
    if(flag){
        int q,amt;
        cout<<"Congratulations!! We have the required book in stock.\n";
        arr[i].details();
        cout<<"Enter the Number of Books you want? \t";
        cin>>q;
        amt=arr[i].query(q);
        if(amt!=0){
            add_to_array(bought[n],arr[i],q);
            cout<<"Cost of books: "<<amt<<endl;
            n++;
        }
        else{
            cout<<"Stock Unavailable.\n";
            flag=false;
        }
        break;
    }
    if(!flag) cout<<"We Do not have the book you require\n";
    break;
}

case 2:
    show_all_books(arr);
    break;

case 3:
    cout<<"Enter the Title and Author of Book\n";
    cin.clear();
    cin.sync();

```

```
getline(cin,title);
getline(cin,author);
flag=false;
for(int i=0;i<M;i++){
    if(arr[i].match(title,author)){
        float price=0;
        flag=true;
        cout<<"Enter new price: ";
        cin>>price;
        arr[i].update_price(price);
        break;
    }
}
if(!flag){
    cout<<"Book not found"<<endl;
}
break;

case 4:
cout<<"Enter Name of Author: ";
cin.clear();
cin.sync();
getline(cin,author);
search_by_author(arr,author);
break;

case 5:
cout<<"Enter Title of Book: ";
cin.clear();
cin.sync();
getline(cin,title);
search_by_title(arr,title);
break;

default:
```

```
cout<<"Enter valid input\n";
break;
}
if(choice==0) break;
}
bill(bought,n);
return 0;
}
```

Output:

D:\sem-5\oop_Lab\6.exe

Enter Details of books in stock.

Details of book 1

Enter Title

DS and Algo

Enter Author's name

Tanmay Vig

Enter Publisher

TV Corp

Enter Price

1000

Enter Stock Position

30

Details of book 2

Enter Title

Basics of ML

Enter Author's name

Tanmay Vig

Enter Publisher

bmp

Enter Price

550

Enter Stock Position

15

Details of book 3

Enter Title

DS and Algo

Enter Author's name

Chandu

Enter Publisher

Carl

Enter Price

300

Enter Stock Position

20

Details of book 4

Enter Title

Atomic Habits

Enter Author's name

James Clear

Enter Publisher

rh

Enter Price

500

Enter Stock Position

10

Details of book 5

Enter Title

Harry Potter

Enter Author's name

JK



D:\sem-5\oop_Lab\6.exe

Enter Stock Position

10

Details of book 5

Enter Title

Harry Potter

Enter Author's name

JK

Enter Publisher

Rowling

Enter Price

250

Enter Stock Position

9

Enter 1 to buy Book

Enter 2 to see all book details

Enter 3 to update price of a book

Enter 4 to search all books by an author

Enter 5 to search books by Title

Enter 0 to exit.

2

Title	Author	Publisher	Price	Stock Position
DS and Algo	Tanmay Vig	TV Corp	1000	30
Basics of ML	Tanmay Vig	bmp	550	15
DS and Algo	Chandu	Carl	300	20
Atomic Habits	James Clear	rh	500	10
Harry Potter	JK	Rowling	250	9

Enter 1 to buy Book

Enter 2 to see all book details

Enter 3 to update price of a book

Enter 4 to search all books by an author

Enter 5 to search books by Title

Enter 0 to exit.

3

Enter the Title and Author of Book

Harry Potter

JK

Enter new price: 500

Enter 1 to buy Book

Enter 2 to see all book details

Enter 3 to update price of a book

Enter 4 to search all books by an author

Enter 5 to search books by Title

Enter 0 to exit.



D:\sem-5\oop_Lab\6.exe

Enter the Title and Author of the book

Atomic Habits

James Clear

Congratulations!! We have the required book in stock.

Title: Atomic Habits

Author: James Clear

Publisher: rh

Price: 500

Enter the Number of Books you want? 9

Cost of books: 4500

Enter 1 to buy Book

Enter 2 to see all book details

Enter 3 to update price of a book

Enter 4 to search all books by an author

Enter 5 to search books by Title

Enter 0 to exit.

2

Title	Author	Publisher	Price	Stock Position
DS and Algo	Tanmay Vig	TV Corp	1000	20
Basics of ML	Tanmay Vig	bmp	550	15
DS and Algo	Chandu	Carl	300	20
Atomic Habits	James Clear	rh	500	1
Harry Potter	JK	Rowling	500	9

Enter 1 to buy Book

Enter 2 to see all book details

Enter 3 to update price of a book

Enter 4 to search all books by an author

Enter 5 to search books by Title

Enter 0 to exit.

4

Enter Name of Author: Tanmay Vig

Title	Publisher	Price	Stock Position
DS and Algo	TV Corp	1000	20
Basics of ML	bmp	550	15

Enter 1 to buy Book

Enter 2 to see all book details

Enter 3 to update price of a book

Enter 4 to search all books by an author

Enter 5 to search books by Title



D:\sem-5\oop_Lab\6.exe

DS and Algo	TV Corp	1000	20	
Basics of ML	bmp	550	15	

Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.

5

Enter Title of Book: DS and Algo

Author	Publisher	Price	Stock Position	
DS and Algo	TV Corp	1000	20	
DS and Algo	Carl	300	20	

Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.

0

Enter valid input

Item: 1

Title: DS and Algo

Author: Tanmay Vig

Price: 1000

Quantity: 10

Item: 2

Title: Atomic Habits

Author: James Clear

Price: 500

Quantity: 9

Total: 14500

Process exited after 339.5 seconds with return value 0

Press any key to continue . . .