Digital Assignment-1

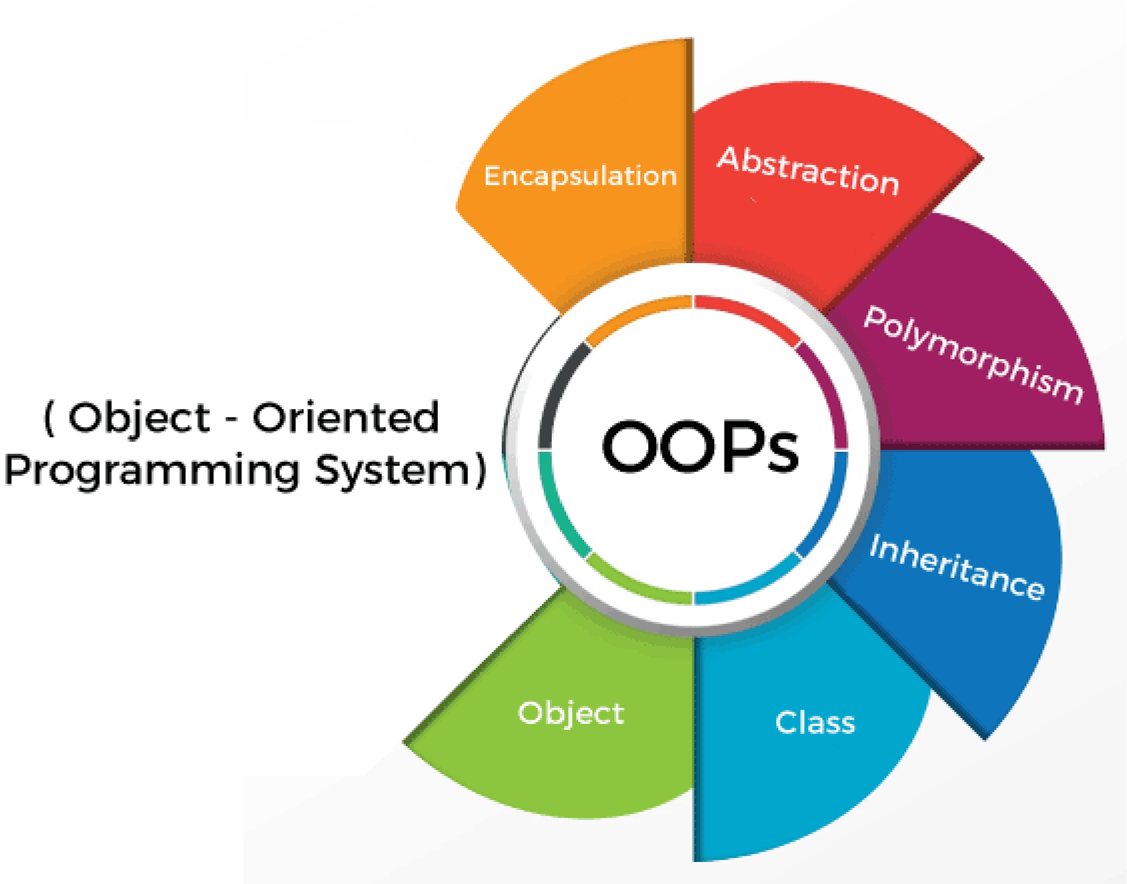
Object Oriented Programming (Lab)

Submitted By: Hari Krishna Shah

VIT ID: 21BCS0167

Google Drive Link:

https://drive.google.com/drive/folders/1QQxJ9Y0CKSeKKXRkVQ-R92jdA3bDtIqI?usp=sharing



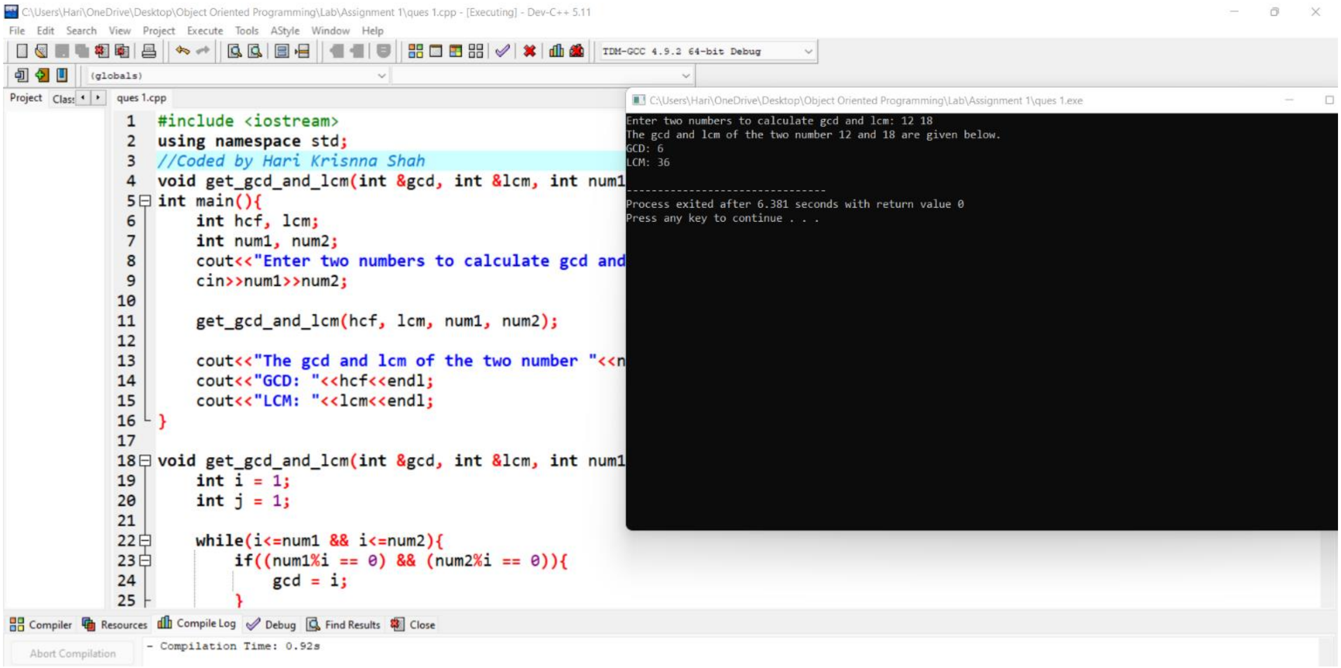
Ques 1. Develop a function that returns through its reference parameters

both the GCD and LCM values passed to the function.

void get_Gcd_Lcm (int &gcd, int &lcm, int num1, int num2);

Answer:

```
#include <iostream>
using namespace std;
//Coded by Hari Krisnna Shah
void get_gcd_and_lcm(int &gcd, int &lcm, int num1, int num2);
int main(){
    int hcf, lcm;
    int num1, num2;
    cout<<"Enter two numbers to calculate gcd and lcm: ";
    cin>>num1>>num2;

    get_gcd_and_lcm(hcf, lcm, num1, num2);

    cout<<"The gcd and lcm of the two number "<<num1<<" and "<<num2<<" are given below."<<endl;
    cout<<"GCD: "<<hcf<<endl;
    cout<<"LCM: "<<lcm<<endl;
}

void get_gcd_and_lcm(int &gcd, int &lcm, int num1, int num2){
    int i = 1;
    int j = 1;

    while(i<=num1 && i<=num2){
        if((num1%i == 0) && (num2%i == 0)){
            gcd = i;
        }
         i += 1;
    }

    j = 1;
    while((j%num1 != 0) || ( j%num2 != 0)){
        j += 1;
    }
    lcm = j;

}
```


C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\Lab\Assignment 1\ques 1.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Debug
(globals)
Project Class
ques 1.cpp
C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\Lab\Assignment 1\ques 1.exe
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation

```
#include <iostream>
using namespace std;
//Coded by Hari Krisnna Shah
void get_gcd_and_lcm(int &gcd, int &lcm, int num1
int main(){
    int hcf, lcm;
    int num1, num2;
    cout<<"Enter two numbers to calculate gcd and
    cin>>num1>>num2;

    get_gcd_and_lcm(hcf, lcm, num1, num2);

    cout<<"The gcd and lcm of the two number "<<n
    cout<<"GCD: "<<hcf<<endl;
    cout<<"LCM: "<<lcm<<endl;
}

void get_gcd_and_lcm(int &gcd, int &lcm, int num1
    int i = 1;
    int j = 1;

    while(i<=num1 && i<=num2){
        if((num1%i == 0) && (num2%i == 0)){
            gcd = i;
        }
```

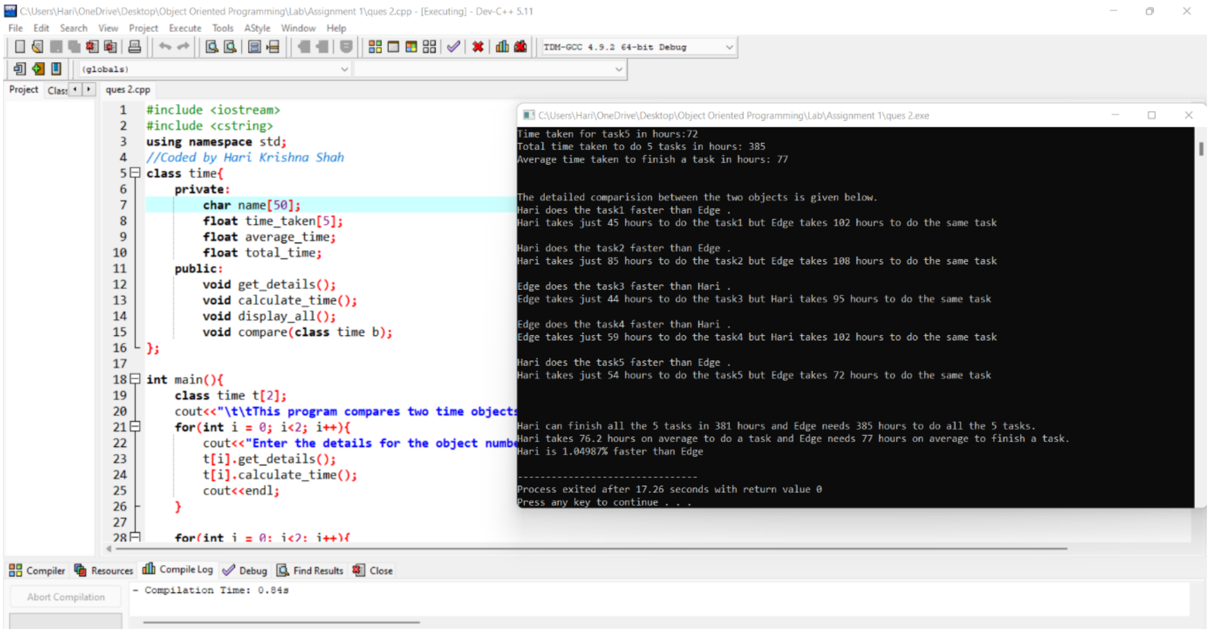
```
Enter two numbers to calculate gcd and lcm: 12 18
The gcd and lcm of the two number 12 and 18 are given below.
GCD: 6
LCM: 36

--------------------------------
Process exited after 6.381 seconds with return value 0
Press any key to continue . . .
```

```
- Compilation Time: 0.92s
```

Ques 2. Develop a program to read the given two time objects which has the time duration taken by the two individuals to complete the set of tasks (Ex.Task1 to Task5). Identify who is efficient in each of the task and in overall all of tasks by comparing the time taken to complete each of the task by individual and the overall time to complete all of the tasks.

Answer:

```cpp
#include <iostream>
#include <cstring>
using namespace std;
//Coded by Hari Krishna Shah
class time{
    private:
        char name[50];
        float time_taken[5];
        float average_time;
        float total_time;
    public:
        void get_details();
        void calculate_time();
        void display_all();
        void compare(class time b);
};

int main(){
    class time t[2];
    cout<<"\t\tThis program compares two time objects and provides statistics based on the comparision."<<endl;
    for(int i = 0; i<2; i++){
        cout<<"Enter the details for the object number "<< i+1 <<" below."<<endl;
        t[i].get_details();
        t[i].calculate_time();
        cout<<endl;
    }

    for(int i = 0; i<2; i++){
        cout<<"The calculated details for the object number "<<i+1<<" is given below."<<endl;
        t[i].display_all();
        cout<<endl<<endl;
    }
    cout<<"The detailed comparision between the two 
objects is given below."<<endl;
    t[0].compare(t[1]);
    return 0;
}

void time::get_details(){
    cout<<"Enter the name for the worker: ";
    cin>>name;
    cout<<"Enter the time take taken by "<<name<<" 
to do the 5 tasks below in terms of hours."<<endl;
    for(int i = 0; i<5; i++){
        cout<<"Time Taken to do task "<<i+1<<" in 
terms of hours: ";
        cin>>time_taken[i];
    }
    cout<<"All details collected 
successfully."<<endl<<endl;
}

void time::calculate_time(){
    float total_temp=0, average_temp;

    for(int i = 0; i<5; i++){
        total_temp += time_taken[i];
    }
    average_temp = total_temp/5;
    total_time = total_temp;
    average_time = total_time/5;
}

void time::display_all(){
    cout<<"The details of time taken by "<<name<<" 
to do 5 tasks are given below."<<endl;
    cout<<"Name: "<<name<<endl;
    for(int i = 0; i<5; i++){
        cout<<"Time taken for task"<<i+1<<" in 
hours:"<<time_taken[i]<<endl;
    }
    cout<<"Total time taken to do 5 tasks in hours:
"<<total_time<<endl;
    cout<<"Average time taken to finish a task in
hours: "<<average_time<<endl;
}

void time::compare(class time b){
    float percent;;
    for(int i = 0; i<5; i++){
        if(time_taken[i]>b.time_taken[i]){
            cout<<b.name<<" does the task"<<i+1<<"
faster than "<<name<<" ."<<endl;
            cout<<b.name<<" takes just
"<<b.time_taken[i]<<" hours to do the task"<<i+1<<"
but "<<name<<" takes "<<time_taken[i]<<" hours to
do the same task"<<endl;
            cout<<endl;
        }
        else if(time_taken[i]<b.time_taken[i]){
            cout<<name<<" does the task"<<i+1<<"
faster than "<<b.name<<" ."<<endl;
            cout<<name<<" takes just
"<<time_taken[i]<<" hours to do the task"<<i+1<<"
but "<<b.name<<" takes "<<b.time_taken[i]<<" hours
to do the same task"<<endl;
            cout<<endl;
        }
        else{
            cout<<"Both "<<name<<" and "<<b.name<<"
take equal time of "<<b.time_taken[i]<<" hours to
do the task"<<i+1<<" ."<<endl;
            cout<<endl;
        }
    }
    cout<<endl<<endl;
    cout<<name<<" can finish all the 5 tasks in
"<<total_time<<" hours and "<<b.name<<" needs
"<<b.total_time<<" hours to do all the 5
tasks."<<endl;
    cout<<name<<" takes "<<average_time<<" hours on
average to do a task and "<<b.name<<" needs
"<<b.average_time<<" hours on average to finish a
task.";
    cout<<endl;
    if(total_time>b.total_time){
        percent = (total_time/b.total_time)*100;
        cout<<b.name<<" is "<<percent-100<<"%
faster than "<<name<<endl;
    }
    else if(b.total_time>total_time){
        percent = (b.total_time/total_time)*100;
        cout<<name<<" is "<<percent-100<<"% faster
than "<<b.name<<endl;
    }
    else{
        cout<<"Both "<<name<<" and "<<b.name<<"
have the same working speed."<<endl;
    }

}
```


C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\Lab\Assignment 1\ques 2.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Debug
(globals)
Project Class
ques 2.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation

```cpp
#include <iostream>
#include <cstring>
using namespace std;
//Coded by Hari Krishna Shah
class time{
	private:
		char name[50];
		float time_taken[5];
		float average_time;
		float total_time;
	public:
		void get_details();
		void calculate_time();
		void display_all();
		void compare(class time b);
};

int main(){
	class time t[2];
	cout<<"\t\tThis program compares two time objects
	for(int i = 0; i<2; i++){
		cout<<"Enter the details for the object numbe
		t[i].get_details();
		t[i].calculate_time();
		cout<<endl;
	}

	for(int i = 0; i<2; i++){
```

C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\Lab\Assignment 1\ques 2.exe

```
Time taken for task5 in hours:72
Total time taken to do 5 tasks in hours: 385
Average time taken to finish a task in hours: 77


The detailed comparision between the two objects is given below.
Hari does the task1 faster than Edge .
Hari takes just 45 hours to do the task1 but Edge takes 102 hours to do the same task

Hari does the task2 faster than Edge .
Hari takes just 85 hours to do the task2 but Edge takes 108 hours to do the same task

Edge does the task3 faster than Hari .
Edge takes just 44 hours to do the task3 but Hari takes 95 hours to do the same task

Edge does the task4 faster than Hari .
Edge takes just 59 hours to do the task4 but Hari takes 102 hours to do the same task

Hari does the task5 faster than Edge .
Hari takes just 54 hours to do the task5 but Edge takes 72 hours to do the same task



Hari can finish all the 5 tasks in 381 hours and Edge needs 385 hours to do all the 5 tasks.
Hari takes 76.2 hours on average to do a task and Edge needs 77 hours on average to finish a task.
Hari is 1.04987% faster than Edge

--------------------------------
Process exited after 17.26 seconds with return value 0
Press any key to continue . . .
```

```
- Compilation Time: 0.84s
```

Ques 3. A book store maintains the inventory of books that are being sold in the store. The list includes the details such as title, author, publisher, price and stock position. Whenever, a customer asks for a book, the sales person inputs the title and author and system searches the list and displays the whether the book is available or not. If the book is available, it displays the total no. of copies available and if it is not available, it displays “STOCK - NOT AVAILABLE - YET TO COME”. Page 2 of 2 If, the requested no. of copies available, total cost of the available copies are displayed. Otherwise, it displays “Required copies are not available”. Develop an OOP using classes and objects. Include necessary constructors and destructor to maintain the objects space and their usage effectively.

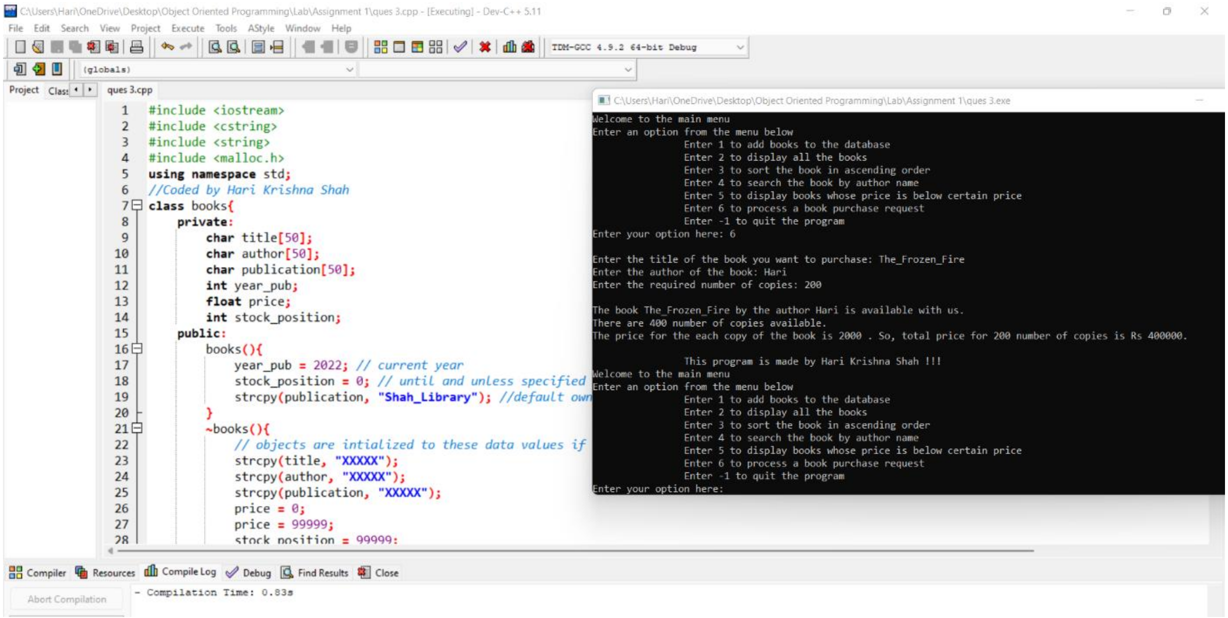
Answer:

```cpp
#include <iostream>
#include <cstring>
#include <string>
#include <malloc.h>
using namespace std;
```

```
//Coded by Hari Krishna Shah
class books{
    private:
        char title[50];
        char author[50];
        char publication[50];
        int year_pub;
        float price;
        int stock_position;
    public:
        books(){
            year_pub = 2022; // current year
            stock_position = 0; // until and unless
specified
            strcpy(publication, "Shah_Library");
//default owner and publisher of the book
        }
        ~books(){
            // objects are intialized to these data
values if they go out of scope. It protects objects
from data leak.
            strcpy(title, "XXXXX");
            strcpy(author, "XXXXX");
            strcpy(publication, "XXXXX");
            price = 0;
            price = 99999;
            stock_position = 99999;
        }
        void get_details();
        string return_title();
        void display();
        string return_author();
        int return_price();
        int request_handle(char book_title[50],
char book_author[50], int required_copies);
        void display_message(int request, int
required_copies);
};
void books::get_details(){
            cout<<"Enter the title of the book: ";
            cin>>title;
            cout<<"Enter the author of the book: ";
            cin>>author;
            cout<<"Enter the publisher of the book:
";
            cin>>publication;
            cout<<"Enter the year of publication of
the book: ";
            cin>>year_pub;
            cout<<"Enter the price of the book: ";
            cin>>price;
            cout<<"Enter the number of copies: ";
            cin>>stock_position;
}
void books::display(){
    cout<<"Book Title: "<<title<<endl;
    cout<<"Author: "<<author<<endl;
    cout<<"Publisher: "<<publication<<endl;
    cout<<"Year of publication: "<<year_pub<<endl;
    cout<<"Price: "<<price<<endl;
    cout<<"Number of copies available:
"<<stock_position<<endl<<endl;
}

string books::return_title(){
    return title;
}
string books::return_author(){
    return author;
}
int books::return_price(){
    return price;
}
```

```
int books::request_handle(char book_title[50], char
book_author[50], int required_copies){
    if((strcmp(book_title, title)==0) &&
((strcmp(book_author, author) ==0)) &&
stock_position >= required_copies){
        return 1;
    }
    else if((strcmp(book_title, title)==0) &&
((strcmp(book_author, author) ==0))){
        return 2;
    }
}

void books::display_message(int request, int
required_copies){
    if(request == 1){
        cout<<"\nThe book "<<title<<" by the author
"<<author<<" is available with us."<<endl;
        cout<<"There are "<<stock_position<<"
number of copies available."<<endl;
        cout<<"The price for the each copy of the
book is "<<price<<" . So, total price for
"<<required_copies<<" number of copies is Rs
"<<price*required_copies<<". "<<endl;
    }
    else if(request == 2){
    cout<<"\nSorry, there are only "
<<stock_position<<" number of copies of the book
"<<title<<" by the author "<<author<<" available
with us."<<endl;
    cout<<"Required number of copies not
available.!!!"<<endl;
    cout<<"We are short by "<<required_copies -
stock_position<<" number of copies."<<endl;
    }
}
```

```
int main(){
    class books *b;
    b = (class books *) (malloc(sizeof(class 
books)));
    static int book_count = 0;
    int temp1_count = 0, temp2_count = 0;
    int option;
    do{
        cout<<"\t\tThis program is made by Hari 
Krishna Shah !!!"<<endl;
        cout<<"Welcome to the main menu"<<endl;
        cout<<"Enter an option from the menu 
below\n \
        Enter 1 to add books to the database\n \
        Enter 2 to display all the books\n \
        Enter 3 to sort the book in ascending 
order\n \
        Enter 4 to search the book by author name\n 
\
        Enter 5 to display books whose price is 
below certain price\n \
        Enter 6 to process a book purchase request 
\n \
        Enter -1 to quit the program"<<endl;
        cout<<"Enter your option here: ";
        cin>>option;
        cout<<endl;
        switch(option){
            case -1: {
                break;
            }
            case 1:{
                cout<<"Enter the number of books 
you want to add: ";
                cin>>temp1_count;
                book_count += temp1_count;
                cout<<endl;
                for(; temp2_count<book_count;
temp2_count++){
                    cout<<"Enter the details for
book number "<<temp2_count+1<<" :"<<endl;
                    b = (class books *)
(realloc(b, book_count*sizeof(class books)));
                    b[temp2_count].get_details();
                    cout<<endl;
                }
                break;
            }
            case 2:{
                if(book_count == 0){
                    cout<<"The database is empty.
Add some books first."<<endl;
                    break;
                }
                for(int i = 0; i<book_count; i++){
                    cout<<"Details for book number
"<<i+1<<" is given below."<<endl;
                    b[i].display();
                    cout<<endl;
                }
                break;
            }
            case 3:{
                if(book_count == 0){
                    cout<<"There's isn't any book
in the database. Add some books to the database
first."<<endl;
                    break;
                }
                class books temp;
                for(int i = 0; i<book_count; i++){
                    for(int j = i+1; j<book_count;
j++){
    if((b[i].return_title().compare(b[j].return_tit
le()))>0){
                            temp = b[i];
                            b[i] = b[j];
                            b[j] = temp;
                        }
                    }
                }
                cout<<"The books have been sorted
in ascending order according to book title
successfully."<<endl;
                break;
            }

            case 4:{
                if(book_count == 0){
                    cout<<"\nThere's isn't any
book in the database. Add some books to the
database first."<<endl;
                    break;
                }
                char search[50];
                cout<<"Enter the author name to
search in the database: ";
                cin>>search;
                int flag1 = 0;
                for(int i = 0; i<book_count; i++){

    if(b[i].return_author().compare(search)==0){
                        if(flag1 == 0){
                            cout<<"\nFollowing
book is authored by "<<search<<"."<<endl;
                        }
                        cout<<"Book number:
"<<flag1+1<<"."<<endl;
                        b[i].display();
                        cout<<endl;
                        flag1 += 1;
                    }
                }
                if(flag1 == 0){
                    cout<<"There isn't any book
authored by "<<search<<" in the database.\nTry
again with another author name."<<endl;
                }
                else{
                    cout<<"Total "<<flag1<<"
numbers of books were found in the
database."<<endl;
                }
                break;
            }

            case 5:
            {
                if(book_count == 0){
                    cout<<"There's isn't any book
in the database. Add some books to the database
first."<<endl;
                    break;
                }
                int price;
                cout<<"Enter the price to find
books below that price: ";
                cin>>price;
                int flag = 0;
                for(int i = 0; i<book_count; i++){
                    if(b[i].return_price()<price){
                        cout<<"Match Found. Book
number: "<<flag+1<<endl;
                        b[i].display();
                        flag += 1;
                    }
                }
                if(flag == 0){
                    cout<<"There isn't any book 
below Rs."<<price<<" in the database."<<endl;
                }
                break;
            }
            case 6:{
                if(book_count == 0){
                    cout<<"There book database is 
empty. Add some books first."<<endl;
                }
                else{
                    char book_title[50], 
book_author[50];
                    int required_copies, flag = 0, 
result=0;
                    cout<<"Enter the title of the 
book you want to purchase: ";
                    cin>>book_title;
                    cout<<"Enter the author of the 
book: ";
                    cin>>book_author;
                    cout<<"Enter the required 
number of copies: ";
                    cin>>required_copies;
                    for(int i = 0; i<book_count; 
i++){
                        result = 
b[i].request_handle(book_title, book_author, 
required_copies);
                        if(result == 1 || result 
== 2){

    b[i].display_message(result, required_copies);
                            flag += 1;
                            break;
                        }
                    }
                    if(flag == 0){
                        cout<<"\nSTOCK- NOT
Available. YET TO COME."<<endl;
                    }
            }
                break;
            }
            default:{
                cout<<"Enter a valid option and
try again."<<endl;
                break;
            }

        }
        cout<<endl;
    }
    while(option != -1);
    cout<<"Thank you for using the program.\nPlease
drop a feedback or a comment."<<endl;

    return 0;
}
```


C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\Lab\Assignment 1\ques 3.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Debug
(globals)
Project Class
ques 3.cpp
C:\Users\Hari\OneDrive\Desktop\Object Oriented Programming\Lab\Assignment 1\ques 3.exe
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation

```cpp
#include <iostream>
#include <cstring>
#include <string>
#include <malloc.h>
using namespace std;
//Coded by Hari Krishna Shah
class books{
    private:
        char title[50];
        char author[50];
        char publication[50];
        int year_pub;
        float price;
        int stock_position;
    public:
        books(){
            year_pub = 2022; // current year
            stock_position = 0; // until and unless specified
            strcpy(publication, "Shah_Library"); //default own
        }
        ~books(){
            // objects are intialized to these data values if
            strcpy(title, "XXXXX");
            strcpy(author, "XXXXX");
            strcpy(publication, "XXXXX");
            price = 0;
            price = 99999;
            stock_position = 99999;
```

```
Welcome to the main menu
Enter an option from the menu below
                Enter 1 to add books to the database
                Enter 2 to display all the books
                Enter 3 to sort the book in ascending order
                Enter 4 to search the book by author name
                Enter 5 to display books whose price is below certain price
                Enter 6 to process a book purchase request
                Enter -1 to quit the program
Enter your option here: 6

Enter the title of the book you want to purchase: The_Frozen_Fire
Enter the author of the book: Hari
Enter the required number of copies: 200

The book The_Frozen_Fire by the author Hari is available with us.
There are 400 number of copies available.
The price for the each copy of the book is 2000 . So, total price for 200 number of copies is Rs 400000.

                This program is made by Hari Krishna Shah !!!
Welcome to the main menu
Enter an option from the menu below
                Enter 1 to add books to the database
                Enter 2 to display all the books
                Enter 3 to sort the book in ascending order
                Enter 4 to search the book by author name
                Enter 5 to display books whose price is below certain price
                Enter 6 to process a book purchase request
                Enter -1 to quit the program
Enter your option here:
```

```
- Compilation Time: 0.83s
```