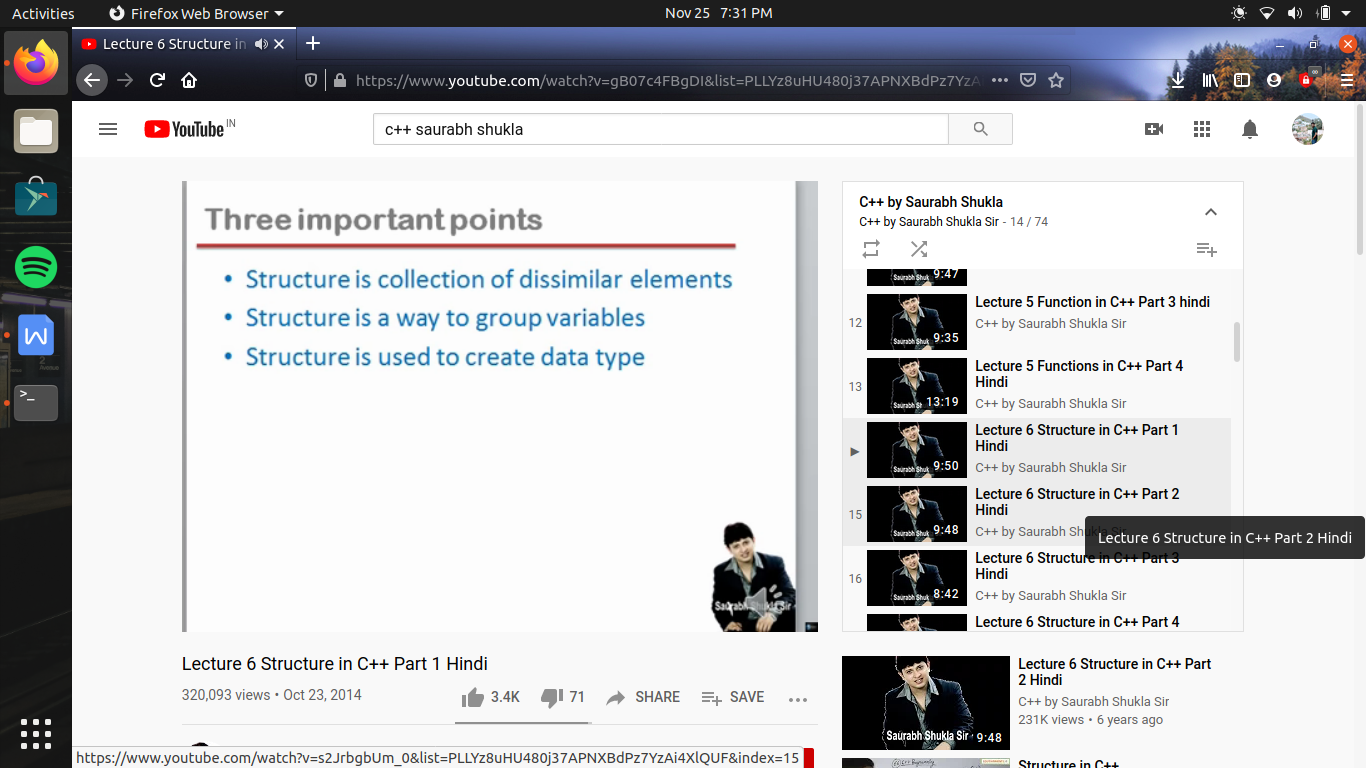
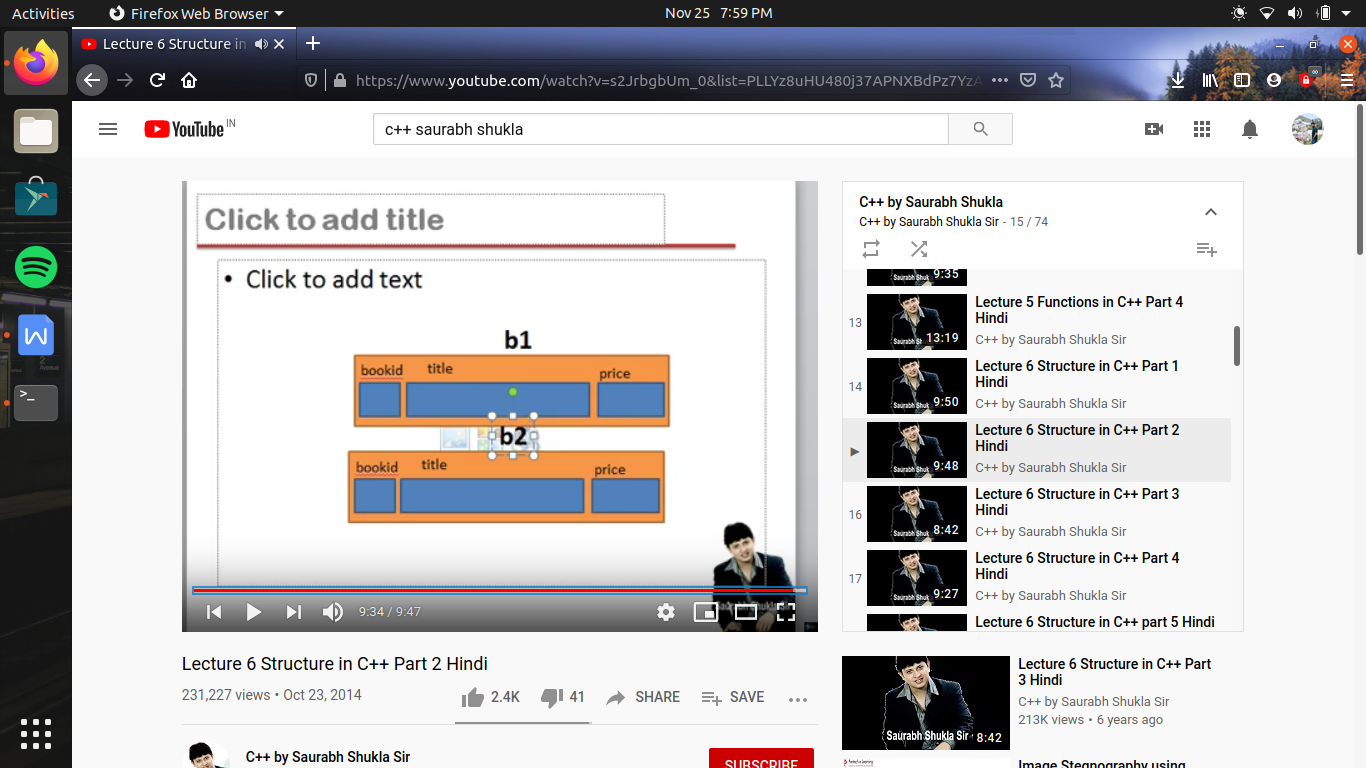
structure in c++



c language me structure ki help se jo bhi data type bana hai agar us ka use kahi bhi karate hai toh us se pahale struct key word likhna compulsory hai

but in c++ struct key word likhana optional hai chahe toh likho warana directly use kr sakate hai wo data type



#include<iostream>

using namespace std;

struct book

{

int book\_id;

char book\_title[40];

float book\_price;

};

int main()

{

book b1={100,"happy songs",45.25}; //here in c++ we don't write struct book

//we initialize the value in sequence

book b2,b3;

b2.book\_id=200;

// b2.title="happy video"; //here b2.title represent the address of first block of array which is a contant so , assignment operator ke left hand side me kabhi constant nhi a sakata thus yeh wrong hai

//toh es ke leye hame pata hona chahiye koe string character array me assign kaise ki jati hai

//es ke leye hum string copy function ka use karate hai strcpy(<address of array>,<string>);

strcpt (b2.title,"happy video");

b2.price = 45.25;

// ab agar hame b2 ka data b3 me rakhana hai toh es ke 2 tarike hai

// pahala ek ek value ko assign karana

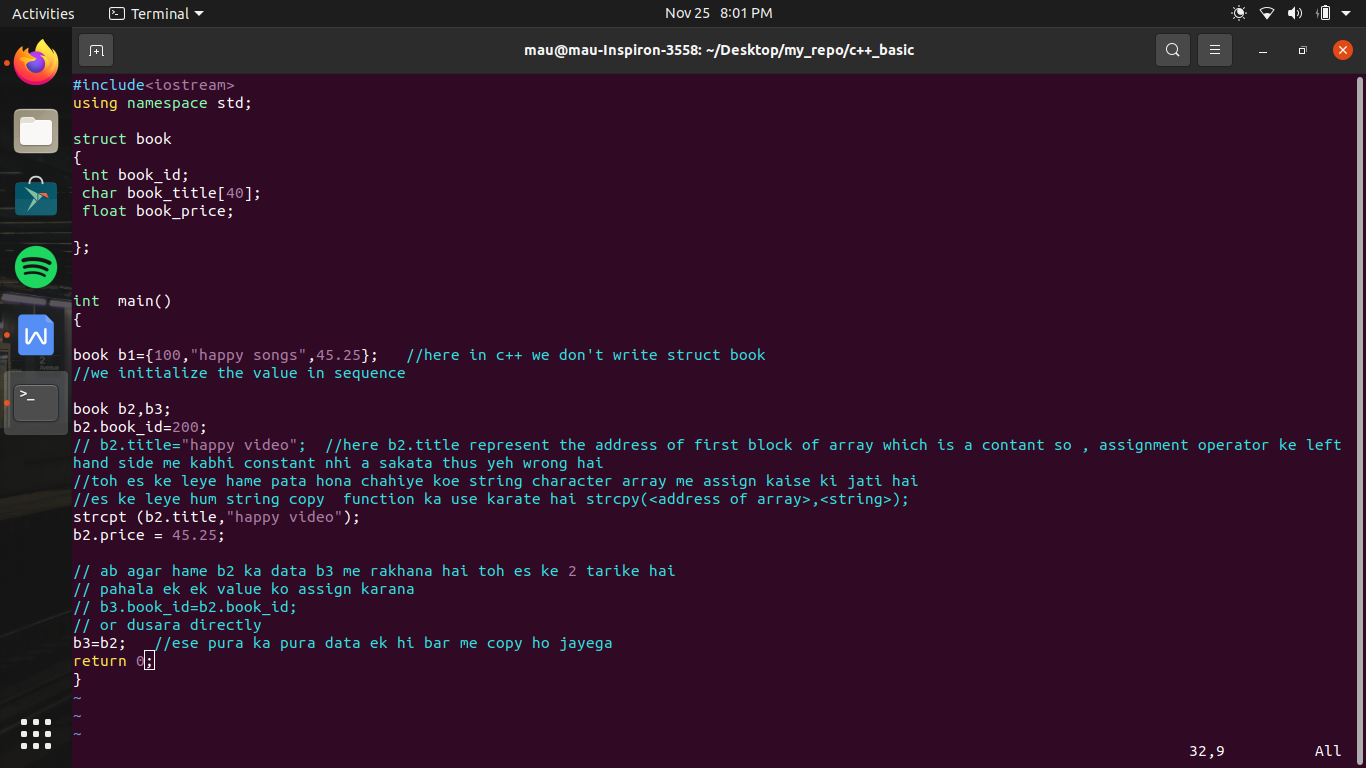
// b3.book\_id=b2.book\_id;

// or dusara directly

b3=b2; //ese pura ka pura data ek hi bar me copy ho jayega

return 0;

}



/\* here we check how to take the data from user \*/

#include<iostream>

using namespace std;

struct book

{

int book\_id;

char book\_title[40];

float book\_price;

};

/\*

int main()

{

book b1;

cout<<"enter book id , title and price of book"<<endl;

cin>>b1.book\_id>>b1.title>>b1.price;

}

\*/ // above program is correct and we also take input by making a function

book input();

void display(book );

int main()

{

book b1;

b1=input();

display(b1);

return 0;

}

book input()

{

book a;

cout<<"enter book id , title and price of book"<<endl;

cin>>a.book\_id>>a.book\_title>>a.book\_price; //here it works as scanf() that means , enter ,space is a delimeter

//thus we can write two words separated by space

return a;

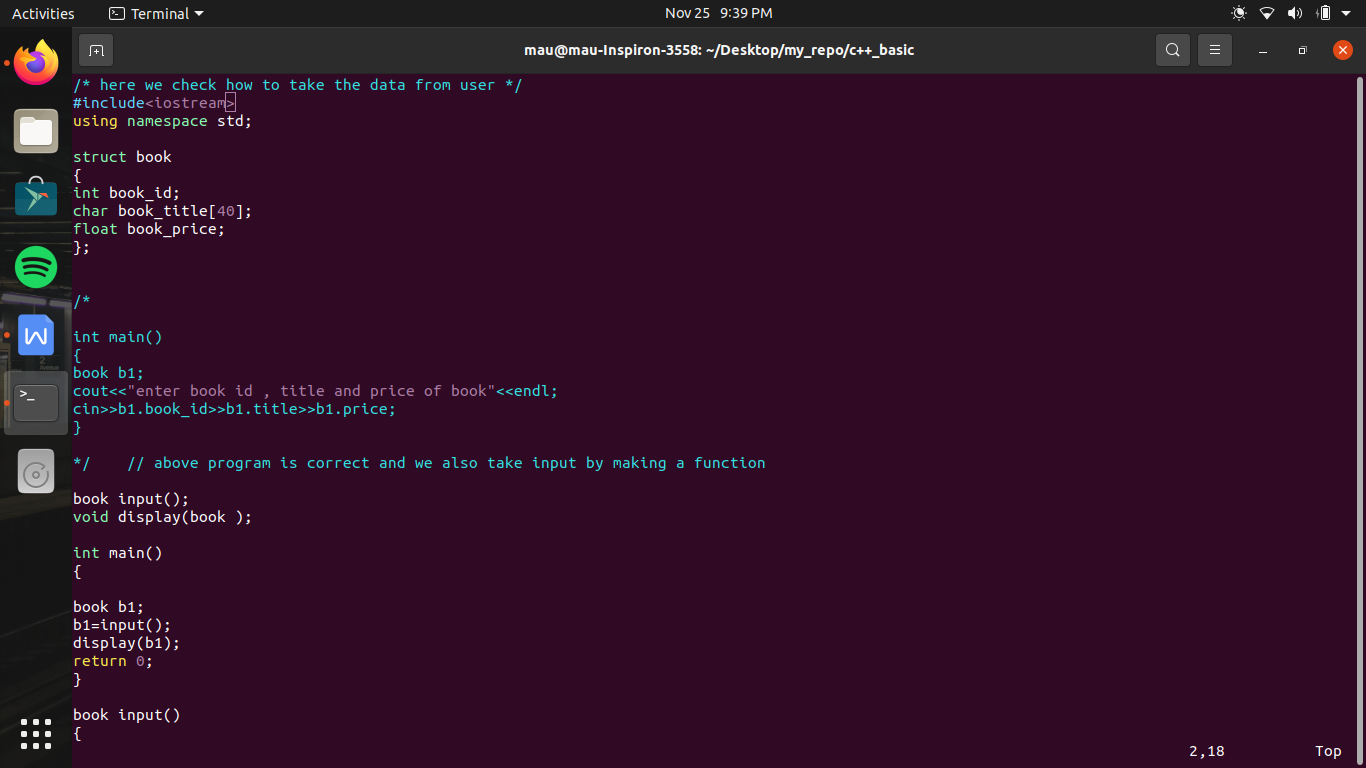
}

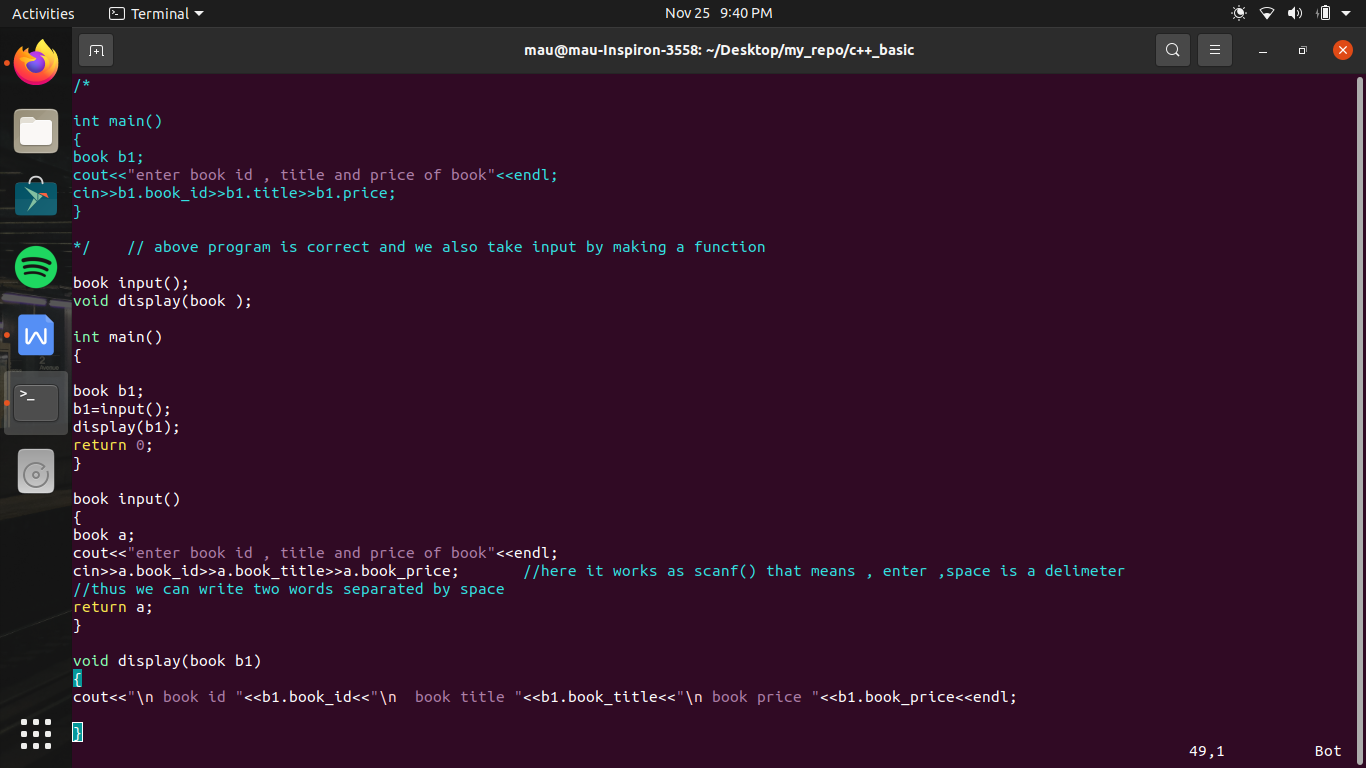
void display(book b1)

{

cout<<"\n book id "<<b1.book\_id<<"\n book title "<<b1.book\_title<<"\n book price "<<b1.book\_price<<endl;

}





OOPs me 5 key principle :-

1. Encapsulation :-

incapsulation yeh kahata hai ki ek entity se related sari properties or sare method ka a group banana chahiye

for example we make a data type book

toh group me yeh sare variable or us variable ko use karane wale function ko ek group me rakhana jaruri hai

aisa karana es leye jaruri hai ki jab hum bada sa program banayege or us me bahot sare structure bane ho alag alag or bahot sare function bane ho toh yeh dhyan me rakhan bahot muskil hoga ki kon sa structure hai jo ek particular structure ko use kr rahe ho , or kon kon se function hai jo dusare structure ko use kar rahe hai

toh funtion basically group me hona chahiye tha jo basically c me possible nhi tha , toh c language me jo structure banate hai us me sirf variables ho sakate hai ,

c++ me hum variable ke alawa functions bhi rkh sakate hai

so as compared to previous program , here now we made display function inside the structure

/\* here we use the first principle of OOPs i.e encapculation \*/

#include<iostream>

using namespace std;

struct book

{

int book\_id;

char book\_title[40];

float book\_price;

void input () //agar hum function ko bhi variables ke sath structure ke andar banayege toh yeh bhi varibables ki tarah member function kahalayega , jaise book\_id, book\_price,book\_title member variables hai , input member function kahalayega

//or member function ko kuch khas adhikar mile hai ki yeh us me se pahala yeh hai ki yeh apane hi structure ke koe bhi member ko yeh directly assess kr sakata hai bina Dot operator ke //i.e why we don't apply dot operator in this

{

cout<<"enter book id , title and price of book"<<endl;

cin>>book\_id>>book\_title>>book\_price;

}

void display() //here we don't need to pass anything we directly access the value

{

cout<<"\n book id "<<book\_id<<"\n book title "<<book\_title<<"\n book price "<<book\_price<<endl;

}

};

int main()

{

book b1,b2;

b1.input(); // ab jaisa ki dhek rahe hai ki input or display function bhi ab member function toh ese access karane keye bhi dot operator lagana padega

b1.display();

//as here b1 is just like a man wkich perform input function i.e kyu ki b1 input function ko call kr raha hai toh jo bhi data input karege jab b1 call krega toh wo data sirf b1 ko belong karega

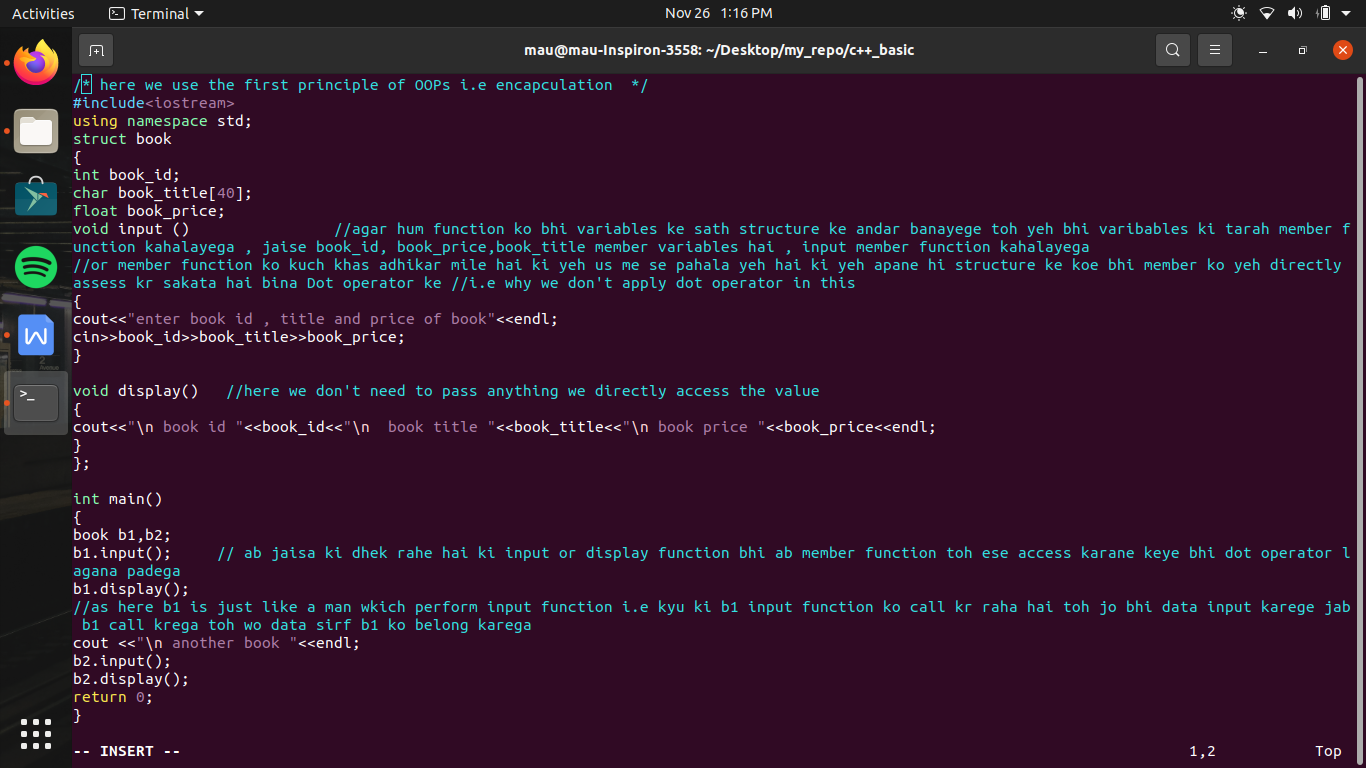
cout <<"\n another book "<<endl;

b2.input();

b2.display();

return 0;

}



ab 3rd difference samajhane ke leye , hum data security ke baharatin concept ko c++ me samajhane ja rahe hai

jaise kuch policy ho sakati hai ki structure se bana kisi variable ki value negative nhi ho sakati hai

jaise hamare example me book\_id ki value kabhi negative nhi ho sakati hai

toh us ke leye hum input function me thoda change karate hai

#include<iostream>

using namespace std;

struct book

{

private:

int book\_id;

char book\_title[40];

float book\_price;

public :

void input ()

{

cout<<"enter book id , title and price of book"<<endl;

cin>>book\_id>>book\_title>>book\_price;

if (book\_id<0) //ese me yeh bola gaya hai ki book id ki value kabhi negative nhi hogi

book\_id=-book\_id;

}

//koe bhi value member variables ke andar kaya ho sakati hai jo ki valid rahe policy ke against na ho

// toh es ki jankari structure banane wale ko hogi toh use yeh dhayan me rakhana hoga hai en variable ki value kabhi corrupt na ho

// c language me aisa ko tarika nhi tha jis se data ki security ki ja sake or use corrupt hone se bachaya ja sake

// agar ek bada program c language me banayege or team me kam hoga toh bahot chances hai ki ek log ko toh malum hoga ki book\_id ki kya value hai or kya honi chahiye but dusare programmer ko nhi malum

// toh us ne kuch galat coding kr di or book id ke andar jo value ani thi wo galat ho gayi

// toh yaha pr bhi aisa ho sakata tha agar hum if condition nhi lagate toh

// but agar main banane wale ko nhi malum ki structure me kya hai or us ne input () function use karane ke bajaye , diretly likh diya ki b1.input = -100;

// toh ab hum ese corrupt hone se nhi bacha pa rahe

// toh es ke leye hum yeh karege ki struture variable ko , structure body ke bahar se koe bhi use hi na kr paye

// yani koe agar likhe b1.book\_id toh error a jaye , or agar aisa kr paye hum toh data ko secure kr lege

// i.e variables ke andar value sirf input function se hi jayegi

// c language me koe tarika nhi tha es galati ko rokane ka but c++ kahati hai ek tarika hai

// toh hum structure body ke andar ek access specifier laga dete hai

// jo ki 3 tarike ke hote hai private , protected , public

// toh agar private likh du toh es ka mtlb hoga ki structure ke body ke bahar se access nhi ho payege keval uske andar se hi access ho payege

// but agar ek bar private specifer likh diya toh us ke bad sare members private ho jayege input function bhi private ho jayega toh

// us me ke or access specifier laga dege , public , jo member public hote hai wo bahar se access ho sakate hai

//

void display()

{

cout<<"\n book id "<<book\_id<<"\n book title "<<book\_title<<"\n book price "<<book\_price<<endl;

}

};

int main()

{

book b1;

b1.input();

b1.display();

return 0;

}

