Static dsa

#include <iostream>

using namespace std;

#include"Book.h"

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main() {

Book b(101,"Prachiti",405.5,"Thakur");

cout<<Book::getcount();

return 0;

}

#include<iostream>

using namespace std;

#include"Book.h"

//define static variable

int Book::count=0;

Book::Book()

{

Book :: count++;

this->bid=0;

strcpy(this->bname,"Book");

this->price=0;

strcpy(this->authour,"XYZ");

}

Book::Book(int i,char\* nm,double p, char\* ath){

Book :: count++;

this->bid=0;

strcpy(this->bname,nm);

this->price=0;

strcpy(this->authour,ath);

}

void Book::setbid(int i){

this->bid=i;

}

void Book::setname(char\* nm)

{

strcpy(this->bname,nm);

}

void Book::setprice(double p){

this->price=p;

}

void Book::setauthour(char\* ath){

strcpy(this->authour,ath);

}

int Book::getbid(){

return this->bid;

}

char\* Book::getbname(){

return this->bname;

}

double Book::getprice(){

return this->price;

}

char\* Book::getauthour(){

return this->authour;

}

//for static members

int Book::getcount(){

return Book :: count;

}

#include<iostream>

using namespace std;

class Book{

int bid;

char bname[20];

double price;

char authour[20];

static int count;

public:

Book();

Book(int ,char\*,double, char\*);

void setbid(int);

void setname(char\*);

void setprice(double);

void setauthour(char\*);

int getbid();

char\* getbname();

double getprice();

char\* getauthour();

//for static members

static int getcount();

};

#include"product.h"

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main() {

Product p(101,"product",8900,90);

double price=p.applyDis();

cout<<"price:"<<price;

return 0;

}

#include<iostream>

using namespace std;

class Product{

int pid;

char pname[20];

double price;

int quantity;

static double discount;

public:

Product();

Product(int,char\*,double,int);

int getId();

char\* getName();

double getPrice();

int getQuantity();

void setId(int);

void setName(char\*);

void setPrice(double);

void setQuantity(int);

~Product();

static int getdiscount();

double applyDis();

};

#include"product.h"

double Product::discount=0.1;

Product::Product(){

this->pid=0;

strcpy(this->pname,"Product");

this->price=0;

this->quantity=0;

}

Product::Product(int i,char\* nm,double p,int q){

this->pid=i;

strcpy(this->pname,nm);

this->price=p;

this->quantity=q;

}

int Product::getId(){

return this->pid;

}

char\* Product::getName(){

return this->pname;

}

double Product::getPrice(){

return this->price;

}

int Product::getQuantity(){

return this->quantity;

}

void Product::setId(int i) {

this->pid=i;

}

void Product::setName(char\* nm){

strcpy(this->pname,nm);

}

void Product::setPrice(double p){

this->price=p;

}

void Product::setQuantity(int q){

this->quantity=q;

}

Product::~Product(){

cout<<"\ndestructor gets called!!";

}

int Product::getdiscount(){

return Product::discount;

}

double Product::applyDis(){

this->price=this->price-(this->price\*discount);

}

#include "shirt.h"

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char\*\* argv) {

Shirt s1(101,"prachiti",'n',1000,'x');

cout<<s1.Discount();

return 0;

}

#include<iostream>

using namespace std;

class Shirt{

int sid;

char sname[20];

char type;//f->formal c->casual

double price;

char size;//s->small,m->medium,l->large ,xl->extra large

static double change;

public:

Shirt();

Shirt(int,char\*,char,double,char);

void setId(int);

void setSname(char\*);

void setType(char);

void setPrice(double);

void setSize(char);

int getId();

char\* getName();

char getType();

double getPrice();

char getSize();

double Discount();

~Shirt();

};

#include"shirt.h"

double Shirt::change=0.1;

Shirt::Shirt(){

this->sid=0;

strcpy(this->sname,"Shirt");

this->price=0;

this->size='\0';

this->type='\0';

}

Shirt::Shirt(int i,char\* nm,char t,double p,char s){

this->sid=i;

strcpy(this->sname,nm);

this->type=t;

this->price=p;

this->size=s;

}

void Shirt::setId(int s){

this->sid=s;

}

void Shirt::setSname(char\* nm){

strcpy(this->sname,nm);

}

void Shirt::setType(char t){

this->type=t;

}

void Shirt::setPrice(double p){

this->price=p;

}

void Shirt::setSize(char s){

this->size=s;

}

int Shirt::getId(){

return this->sid;

}

char\* Shirt::getName(){

return this->sname;

}

char Shirt::getType(){

return this->type;

}

double Shirt::getPrice(){

return this->price;

}

char Shirt::getSize(){

return this->size;

}

double Shirt::Discount(){

if(this->size=='s'||this->size=='S'){

this->price=this->price+(this->price\*change\*0);

}

else{

if(this->size=='m'||this->size=='M'){

this->price=this->price+(this->price\*change\*1);

}

else{

if(this->size=='l'||this->size=='L'){

this->price=this->price+(this->price\*change\*2);

}

else{

if(this->size=='x'||this->size=='X'){

this->price=this->price+(this->price\*change\*3);

}

}

}

}

return this->price;

}

Shirt::~Shirt()

{

cout<<"\nDestructor gets called!!";

}