Follow Coding Standards

Day 6 Assignments

1. Create the following classes.

class time {

int hr,min,sec;

public:

time(int ,int,int);

time(const time&) //copy constructor

void increment();

void get(int&,int&,int&)const; //

};

class clock{

time curTime;

public:

clock ( int ,int ,int );

clock ( time);

void dispCurTime()const;

void incrTime();

};

(Note: Both classes do not have any default constructor)

Test the classes with the following main

void main(){

clock c1( 22,45,50);

time t1(9,30,20);

clock c2(t1);

c1.dispCurTime();

c2. dispCurTime();

for(int I = 0; I< 1000 ;I++){

c1.incrTime();

}

for(I = 0; I< 500 ;I++){

c2.incrTime();

}

c1.dispCurTime();

c2. dispCurTime();

}

1. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, and price and stock(int) 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 not, an appropriate message is displayed. If the requested copies are available, display “book is available” and decrement the count. Otherwise the message "Required copies not in stock" is displayed.

Design system using a class called book with suitable member function and constructors. The member variables author, title, price are private members .Use new operator in constructors to allocate memory space required for author, title.

Implement the following menu by declaring array of pointers of CBook type. Add member functions to perform the following

a. Add book : Create a book dynamically and enter all the details including no: of such books.

b. Search book //Just return true/false telling whether book is available or not. Pass author and title.

c. Display stock //Display details of books available in the stock.

d. Delete book. // If author and title is given as parameters , such a book must be deleted(decrement the stock).

1. Implement a Stack class as given below

#define SIZE 5

class CStack{

int\* number;

int position;

public:

CStack();

~CStack();

void Push(int n);

void Pop();

int IsEmpty();

int IsFull();

};

Use the following main function

void main(){

CStack stack;

int m\_nCh;

int m\_nNum;

for (;;) {

cout<< "1.Push\n2 Pop\n3 Exit\nEnter Choice:";

cin>> m\_nCh;

switch (m\_nCh){

case 1:

cout<<"Enter the number\n";

cin>>m\_nNum;

stack.Push(m\_nNum);

break;

case 2:

stack.Pop();

break;

case 3:

exit(1);

}

}

}

**--------------------------------------**