**19F0228**

# Muhammad zain

# Cs(e)

Task 1

# Header File:

## BOOK.H

#include<iostream>

#include<string>

#include<cstdlib>

#include<ctime>

#include<iomanip>

using namespace std;

struct date {

int day;

int month;

int year;

};

class kitab

{

private:

int Id;

string Name;

string Author;

int Edition;

date Date;

public:

kitab();

void input();

void print();

};

# HEADER.CPP:

## BOOK.CPP

#include<iostream>

#include<string>

#include<cstdlib>

#include<ctime>

#include<iomanip>

#include"book.h"

using namespace std;

kitab::kitab()

{

srand((0));

Id = rand() % 1000;

Name = "Default";

Author = "Default";

Edition = 1;

Date.day = 0; Date.month = 0; Date.year = 0;

}

void kitab::input()

{

cout << "Enter Name of the book" << endl;

cin >> Name;

cout << "Enter Author Name" << endl;

cin >> Author;

cout << "Enter Edition" << endl;

cin >> Edition;

cout << "Enter date of Publication" << endl;

cin >> Date.day >> Date.month >> Date.year;

}

void kitab::print()

{

cout << Name << setw(20) << Author << setw(20) << Id << setw(20) << Edition << "th" << setw(20) << Date.day << "-" << Date.month << "-" << Date.year << endl;

}

# DRIVERBOOK:

#include<iostream>

#include<string>

#include<cstdlib>

#include<ctime>

#include<iomanip>

#include"book.h"

using namespace std;

int main()

{

kitab \*objptr;

kitab b;

objptr = &b;

int numberofbooks;

cout << "How many books you want to enter" << endl;

cin >> numberofbooks;

objptr = new kitab[numberofbooks];

cout << endl;

for (int count = 0; count<numberofbooks; count++)

{

objptr[count].input();

system("cls");

cout << "Enter the data of second book" << endl;

}

system("cls");

cout << "Name" << setw(20) << "Author" << setw(20) << "ID" << setw(20) << "Edition" << setw(20) << "DateofPublish" << endl << endl;

for (int count = 0; count<numberofbooks; count++)

{

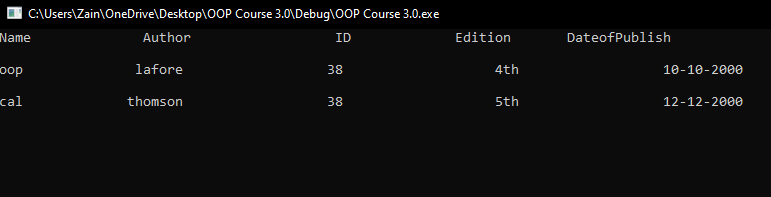
objptr[count].print();

cout << endl;

}

system("pause>0");}

# Snip;



Task 3☹ incomplete

# Header

## Employee.h

#include<iostream>

#include<string>

#include<cstdlib>

using namespace std;

class Employee{

private:

int id;// (constant)

string name;

int salary;

int salaryPerHour;

int monthlysalary;

int changesalary;

int monthlyWorkedHours;

int taxPaid;

int yearofJoining;

public :

void set\_id(int );

void set\_name(string);

void set\_salaryPerHour(int );

void set\_monthlyWorkedHours(int);

void updatehours(int);

void set\_monthlysalary(int);

void set\_changesalary(int);

void set\_taxPaid(int);

void set\_yearofJoining(int,int,int);

void minus\_Healthinsurance(int);//1000

void bonus(int);//if a person is working for more than 8 years

void Receipt();

void input();

//default constructor

Employee(){

id = 000;

name=" ";

salaryPerHour=000;

monthlyWorkedHours=000;

salary=000;

taxPaid=000;

yearofJoining=000;

}

};

# Employee.cpp

void Employee::set\_id(int Id)

{

id=Id;

}

void Employee::set\_name(string Name)

{

name=Name;

}

void Employee::set\_salaryPerHour(int SalaryPerHour)

{

salaryPerHour=SalaryPerHour;

}

void Employee::set\_monthlyWorkedHours(int MonthlyWorkedHours)

{

monthlyWorkedHours=MonthlyWorkedHours;

}

void Employee:: set\_monthlysalary(int Monthlysalary)

{

monthlysalary = Monthlysalary;

}

void Employee:: set\_changesalary(int Changesalary)

{

changesalary=Changesalary;

}

int Employee::set\_yearofJoining{

cout<<"for how many years you are working here"<<endl;

int n;

cin>>n;

return n;

}

void Employee:: minus\_Healthinsurance(){

cout<<"From your salary 1ooo has been deducted in account of health insurance ";

cout<<endl;

}

Task 4

# Header

#include<iostream>

#include<string>

using namespace std;

class Chat\_Box\_App {

private:

int ID;

string Name;

string Contact\_No;

string txt[10] = { "" };

int counter;

public:

Chat\_Box\_App();

Chat\_Box\_App\* range(Chat\_Box\_App \*pointer, int size);

void User\_Id();

int getUser\_Id();

void User\_Name();

string getUser\_Name();

void User\_Contact();

string getUser\_Contact();

int Chat\_Functions();

void Text (Chat\_Box\_App\* ptr, int size, int dummy);

void Sender (Chat\_Box\_App ptr[],int length);

void Inbox (Chat\_Box\_App\* ptr, int length);

int Memory\_Check (Chat\_Box\_App\* ptr, int length);

void Empty\_Message\_Box (Chat\_Box\_App\* ptr, int length);

void Received\_Messages (Chat\_Box\_App\* ptr, int length);

int search (Chat\_Box\_App\* ptr, int size);

};

# Cpp

#include<iostream>

#include<string>

#include"Chat\_Box\_App.h"

using namespace std;

Chat\_Box\_App::Chat\_Box\_App() {

ID = 1;

}

Chat\_Box\_App\* range(Chat\_Box\_App \*pointer, int size)

{

Chat\_Box\_App \*ptrnew;

ptrnew = new Chat\_Box\_App[size + 1];

for (int i = 0; i < size; i++)

{

ptrnew[i] = pointer[i];

}

delete[] pointer;

return ptrnew;

}

void Chat\_Box\_App::User\_Id() {

cout << "Enter Unique ID: ";

cin >> ID;

}

int Chat\_Box\_App::getUser\_Id()

{

return ID;

}

void Chat\_Box\_App::User\_Name() {

cout << "Enter your Name" << endl;

cin >> Name;

}

string Chat\_Box\_App::getUser\_Name() {

return Name;

}

void Chat\_Box\_App::User\_Contact() {

cout << "Enter Your Contact Number !" << endl;

cin >> Contact\_No;

}

string Chat\_Box\_App::getUser\_Contact() {

return Contact\_No;

}

int Chat\_Box\_App::Chat\_Functions() {

int menu = 0;

cout << "CHAT BOX APPLICATION FUNCTIONALITIES !! " << endl << endl;

cout << "1-SEND A MESSAGE\n2-VIEW A MESSAGE\n3-EMPTY MESSAGE BOX\n4-UPDATE INFO (NAME & CONTACT)\n5-MEMORY CHECK\n\n";

cin >> menu;

return menu;

}

void Chat\_Box\_App::Text(Chat\_Box\_App\* ptr, int size, int dummy) {

int n= 0;

n = ptr[dummy].Memory\_Check(ptr, dummy);

if (n != 0)

{

cout << "Enter Your Message: ";

getline(cin, ptr[dummy].txt[10 - n]);

while (ptr[dummy].txt[10 - n] == "")

{

getline(cin, ptr[dummy].txt[10 - n]);

}

}

else

{

cout << "There's no More Room in Your Reciever's Inbox. Sorry!" << endl;

}

}

void Chat\_Box\_App::Sender(Chat\_Box\_App ptr[], int length) {

counter =counter+ 1;

ptr[length].User\_Id();

ptr[length].User\_Name();

ptr[length].User\_Contact();

}

void Chat\_Box\_App::Inbox(Chat\_Box\_App\* ptr, int length) {

int n = 0;

n = Memory\_Check(ptr, length);

if (n == 10)

{

cout << "There Is No Message In Inbox." << endl;

}

else

{

for (int check = 0; check < (10 - n); check++)

{

cout << "Message " << check + 1 << ": " << endl;

cout << ptr[length].txt[check] << endl;

}

}

}

int Chat\_Box\_App::Memory\_Check(Chat\_Box\_App\* ptr, int length) {

int n = 0;

for (int check = 0; check < 10; check++)

{

if (ptr[length].txt[check] == "")

{

n++;

}

}

return n;

}

void Chat\_Box\_App::Empty\_Message\_Box(Chat\_Box\_App\* ptr, int length) {

for (int check = 0; check < 10; check++)

{

ptr[length].txt[check] = "";

}

}

void Chat\_Box\_App::Received\_Messages(Chat\_Box\_App\* ptr, int length) {

ptr[length].User\_Name();

ptr[length].User\_Contact();

}

int search(Chat\_Box\_App\* pointer, int size)

{ string Contact\_number;

cout << "Enter Contact Number: ";

getline(cin, Contact\_number);

int negative = -1;

int count = 0;

for (int i = 0; i < size; i++)

{

if (pointer[i].getUser\_Contact() == Contact\_number)

{

negative = i;

break;

}

}

if (negative == -1)

{

cout << "incorrect" << endl;

cout << "input again";

search(pointer, size);

}

return negative;

}