Problem No: 1

Problem Name: CSE department wants to manage a Fast-food shop in its premise. Every student can make pre-order of his/her breakfast before 10 PM of the previous day. A sales person can manage the data and sells the pre-ordered item to the students. If a student pre-ordered before but not take his/her breakfast and the sales person can inform it to the department. If he will not be illegible to pre-order the breakfast another time. As a student of CSE, write OOP code for the project. All the communications will be held by message.

Source Code:

#include<bits/stdc++.h>

#include<string.h>

#include<ctime>

#include<fstream>

using namespace std;

class Date

{

public:

string date;

Date(){};

Date(string date)

{

this->date=date;

}

};

class Person

{

public:

string name;

Date date;

Person(){};

Person(string name)

{

this->name=name;

}

Person(string name,string date)

{

this->name=name;

this->date.date=date;

};

};

class Department

{

public:

string dpt\_name;

Department(){};

Department(string name)

{

dpt\_name=name;

}

};

class Student : public Person

{

public:

string id;

Department department;

Student(){};

Student(string name,string date,Department dpt\_name,string id\_no) : Person(name,date)

{

id=id\_no;

department=dpt\_name;

}

view\_student\_info()

{

cout<<"Student name: "<<name<<endl;

cout<<"Department: "<<department.dpt\_name<<endl;

cout<<"Student ID: "<<id<<endl;

cout<<"Students Birthday: "<<this->date.date<<endl;

}

friend orderfood();

friend blacklist();

};

class Sellsman : public Person

{

public:

int seller\_no;

Sellsman(){};

Sellsman(string name,string date,int seller\_no): Person(name,date)

{

this->seller\_no=seller\_no;

};

view\_sellsman()

{

cout<<"Sells person Name: "<<name<<endl;

cout<<"Birthday: "<<this->date.date<<endl;

}

friend orderfood();

};

class Login

{

string user\_name;

string password;

public:

Login(){};

Login(string user\_name,string password)

{

this->user\_name=user\_name;

this->password=password;

}

friend blacklist();

};

class Faculty : public Person

{

public:

string designation;

Department department;

Login login;

Faculty(){};

Faculty(string designation,string name,Department department1,Login \*login2):Person(name)

{

this->designation=designation;

department=department1;

login=\*login2;

}

view\_faculty()

{

cout<<"Name: "<<name<<endl;

cout<<"Department: "<<department.dpt\_name<<endl;

cout<<"Designation: "<<designation<<endl;

}

Blacklist(int order\_data[],string deliver\_data[],Student \*student[],string blacklist[])

{

cout<<"Blacklisted names are: \n";

for(int n=0;n<4;n++)

{

if(order\_data[n]==1)

if(deliver\_data[n]=="Didn't")

{

blacklist[n]=student[n]->name;

cout<<"Name: "<<student[n]->name<<"\nID: "<<student[n]->id<<"\nYou have been blacklisted, Contact with Department"<<endl;

}

}

};

};

int view\_system\_time()

{

time\_t k = time(0);

struct tm \*t = localtime(&k);

cout<< t->tm\_hour << ":" << t->tm\_min << endl;

return t->tm\_hour;

}

int orderfood(Student \*student)

{

int tm=view\_system\_time();

if(tm>22)

{

cout<<"Sorry, You cannot Place the pre-order.\n";

return -1;

}

else

{

cout<<"Name: "<<student->name<<"\nID: "<<student->id<<"\n Your Order Has been placed.\n";

return 1;

}

}

int main()

{

int number\_of\_students=4;

int temp=number\_of\_students;

Department department("CSE");

Student \*students[number\_of\_students]=

{

new Student("Jahangir","22-6-1994",department,"1008"),

new Student("Abid","10-03-1995",department,"1009"),

new Student("Saidul","21-01-1996",department,"1010"),

new Student("Firoz","15-07-1995",department,"1011")

};

Sellsman sells\_person("Nazmul Karim","13-03-1982",1);

sells\_person.view\_sellsman();

Login \*login\_data[2]={

new Login("RAB","111"),

new Login("KAL","222")

};

Faculty \*faculty\_member[2]=

{

new Faculty("Professor","Amanat Ahmed",department,login\_data[0]),

new Faculty("Lecturer","Khairul Islam",department,login\_data[1])

};

for(int i=0;i<3;i++)

students[i]->view\_student\_info();

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

faculty\_member[i]->view\_faculty();

int order\_data[number\_of\_students],g=0;

for(int i=0;i<number\_of\_students;i++)

{

g=orderfood(students[i]);

order\_data[i]=g;

}

string delivery\_info[number\_of\_students]={"ordered","Didn't","ordered","Didn't"};

string blacklisted\_members[number\_of\_students];

faculty\_member[0]->Blacklist(order\_data,delivery\_info,students,blacklisted\_members);

ofstream students\_file;

students\_file.open("Students.csv");

students\_file<<"Students are: \n";

students\_file<<"Roll,Name,Department\n";

for(int i=0;i<number\_of\_students;i++)

{

students\_file<<students[i]->id<<","<<students[i]->name<<","<<students[i]->department.dpt\_name<<endl;

}

ofstream blacklist\_file; ///Blacklist part

blacklist\_file.open("Blacklist.csv");

int counter=1;

blacklist\_file<<"Blacklisted names are:\n";

blacklist\_file<<"Roll,Name,Department\n";

string blacklisted\_id[number\_of\_students];

for(int i=0;i<number\_of\_students;i++)

{

if(blacklisted\_members[i]!="\0")

{

blacklist\_file<<students[i]->id<<","<<blacklisted\_members[i]<<","<<students[i]->department.dpt\_name<<endl;

counter++;

blacklisted\_id[i]=students[i]->id;

}

}

blacklist\_file.close();

cout<<"Blacklisted person included in file.\n";

cout<<"New to order service?\nPress 1 to register or 0 to exit\n"<<endl;

int new\_order;

cin>>new\_order;

while(new\_order!=0)

{

string name,department\_name,birthday,roll;

cout<<"Enter Name: ";

cin>>name;

cout<<"\nEnter ID:";

cin>>roll;

cout<<"\nEnter Department: ";

cin>>department\_name;

cout<<"\nEnter birthday:";

cin>>birthday;

students[number\_of\_students]={new Student(name,birthday,department\_name,roll)};

number\_of\_students+=1;

int order\_data1;

order\_data1=orderfood(students[number\_of\_students-1]);

cout<<"New to order service?\nPress 1 to register or 0 to exit"<<endl;

cin>>new\_order;

if(new\_order!=1)

break;

}

string delivery\_data2[number\_of\_students]={"ordered","Didn't","ordered","Didn't","ordered"};

for(int i=temp;i<number\_of\_students;i++)

{

students\_file<<students[i]->id<<","<<students[i]->name<<","<<students[i]->department.dpt\_name<<endl;

}

for(int i=0;i<number\_of\_students;i++)

{

if(blacklisted\_id[i]!="\0" && students[i]->id==blacklisted\_id[i] )

cout<<students[i]->name<<"\nYou cannot order,You have been blacklisted.\nContact with Department"<<endl;

else

orderfood(students[i]); } }

Sample Output:

Sells person Name: Nazmul Karim

Birthday: 13-03-1982

Student name: Jahangir

Department: CSE

Student ID: 1008

Students Birthday: 22-6-1994

Student name: Abid

Department: CSE

Student ID: 1009

Students Birthday: 10-03-1995

Student name: Saidul

Department: CSE

Student ID: 1010

Students Birthday: 21-01-1996

Name: Amanat Ahmed

Department: CSE

Designation: Professor

Name: Khairul Islam

Department: CSE

Designation: Lecturer

3:31

Name: Jahangir

ID: 1008

Your Order Has been placed.

3:31

Name: Abid

ID: 1009

Your Order Has been placed.

3:31

Name: Saidul

ID: 1010

Your Order Has been placed.

3:31

Name: Firoz

ID: 1011

Your Order Has been placed.

Blacklisted names are:

Name: Abid

ID: 1009

You have been blacklisted, Contact with Department

Name: Firoz

ID: 1011

You have been blacklisted, Contact with Department

Blacklisted person included in file.

New to order service?

Press 1 to register or 0 to exit

Problem No: 2

Problem Name: A Mess owner wants to develop a software for its mess members. Everyday mess member meal details will be entered in the software and after the month it will show the bill of the mess member. Mess member deposit at least 1000 tk at beginning of the month. As a student of CSE, write OOP code for the project. All the communications will be held by message.

Source Code:

#include<bits/stdc++.h>

#include<string>

#include<fstream>

using namespace std;

class Mess {

public:

string name;

Mess() {};

Mess(string a) {

name = a; }

friend view\_Print(); };

class Date {

public:

string date;

Date() {};

Date(string s) {

date=s; };

friend view\_Print(); };

class Person {

public:

string name;

Date date;

Person() {};

Person(string n,Date \*date) {

name=n;

this->date=\*date;

};

friend view\_Print(); };

class Owner : public Person

{

public:

Mess mess;

string date;

Owner() {};

Owner(string name, string date) {

this->name = name;

this->date = date;

}

friend view\_Print(); };

class Member : public Person {

public:

Mess mess;

string date;

double deposit;

Member() {};

Member(string name, string date,double deposit) {

this->name = name;

this->date = date;

this->deposit = deposit; }

friend view\_Print(); };

double Meal\_Rate(double totalmeal,double Cost) {

double meal\_rate=Cost/totalmeal;

return meal\_rate; };

double Balance(int tmeal,double mealrate,double deposit) {

double myMeal;

myMeal=mealrate\*tmeal;

double blnc;

blnc=deposit-myMeal;

return blnc; }

view\_Print(Mess m,Owner o,Member \*x[],int y[],int z[],int p[],int q[],double s,double k[], double w,int e,int f,int g,int h,Date \*d1[]) {

cout << "\nMess Name : " << m.name << endl;

cout << "Mess Owner's Name : " << o.name << endl;

cout << "Date of Birth : " << o.date << endl;

cout << "-----------------------------------------------------" << endl;

cout << "-----------------------------------------------------" << endl;

cout << endl << endl;

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

cout << "Member "<< " " << i+1 << " : " << endl;

cout << "NAME : " << x[i]->name << endl;

cout << "DATE OF BIRTH : " << x[i]->date<< endl;

cout << "---------------------------------------" << endl;

cout << "Deposit : " << x[i]->deposit<< endl;

cout << "Total Morning Meal : " << y[i] <<endl;

cout << "Total Noon Meal : " << z[i] <<endl;

cout << "Total Night Meal : " << p[i] <<endl;

cout << "Total Meal : " << q[i] << endl;

double a = Balance(q[i],s,x[i]->deposit);

cout << "Balance : " << a << endl;

cout << endl << endl; }

cout << "\_\_\_\_\_\_\_\_\_MEAL INFORMATION\_\_\_\_\_\_\_\_\_" << endl << endl << endl;

double TotalDeposit = x[0]->deposit + x[1]->deposit;

cout << "Total Deposit : " << TotalDeposit << endl;

cout << endl << endl << "Daily Shopping Costs--> " << endl ;

for(int i = 0; i<4; i++) {

cout << "Day " << i+1 << "-"<<d1[i]->date << " : " << k[i] << endl; }

cout << endl << endl << "Total Cost : " << w << endl;

cout << "Meal Rate : " << s << endl;

cout << "Total Morning Meal : " << e << endl;

cout << "Total Noon Meal : " << f << endl;

cout << "Total Night Meal : " << g << endl;

cout << "Total Meal : " << h << endl;

double c = Balance(h,s,TotalDeposit);

cout << "Total Balance : " << c << endl; }

int main() {

Mess mess("SARKER VILLA");

Owner owner("SARKER","04/01/1955");

Member \*member[2]= {

new Member("Abid","10/01/1995",1500),

new Member("Adnan","15/2/1996",1400), };

Date \*date[4]= {

new Date("09/10/2018"),

new Date("09/10/2018"),

new Date("24/10/2018"),

new Date("24/10/2019"), };

int day[4][2][3] = {{{1,1,1},{1,1,0}},{{0,1,1},{1,1,1}},{{1,1,1},{1,1,0}},{{1,0,1},{1,1,1}}};

int TotalMeal = 0, TotalMorningMeal = 0, TotalNoonMeal = 0, TotalNightMeal = 0;

for(int p = 0; p<4; p++) {///ToTal Meal;

for(int q = 0; q<2; q++)

{

for(int r = 0; r<3; r++) {

TotalMeal += day[p][q][r]; } } }

for(int p = 0; p<4; p++){ ///ToTal Morning Meal;

for(int q = 0; q<2; q++) {

for(int r = 0; r<1; r++) {

TotalMorningMeal += day[p][q][r]; } } }

for(int p = 0; p<4; p++){ ///ToTal Noon Meal

for(int q = 0; q<2; q++) {

for(int r = 1; r<2; r++) {

TotalNoonMeal += day[p][q][r]; } } }

for(int p = 0; p<4; p++) { ///ToTal Night Meal

for(int q = 0; q<2; q++) {

for(int r = 2; r<3; r++) {

TotalNightMeal += day[p][q][r]; } } }

int Morning1 = 0,Morning2 = 0, Noon1 = 0,Noon2 = 0,Night1 = 0, Night2 = 0;

int TotalMember1Meal = 0, TotalMember2Meal= 0;

for(int p = 0; p<4; p++) {///ToTal MEMBER 1 Morning Meal

for(int q = 0; q<1; q++) {

for(int r = 0; r<1; r++) {

Morning1 += day[p][q][r]; } } }

for(int p = 0; p<4; p++) { ///ToTal MEMBER 2 Morning Meal

for(int q = 1; q<2; q++) {

for(int r = 0; r<1; r++) {

Morning2 += day[p][q][r]; } } }

int Morning[2] = {Morning1,Morning2};

for(int p = 0; p<4; p++) ///ToTal MEMBER 1 Noon Meal {

for(int q = 0; q<1; q++) {

for(int r = 1; r<2; r++) {

Noon1 += day[p][q][r]; } }

for(int p = 0; p<4; p++) ///ToTal MEMBER 2 Noon Meal {

for(int q = 1; q<2; q++) {

for(int r = 1; r<2; r++) {

Noon2 += day[p][q][r]; } }

int Noon[2] = {Noon1,Noon2};

for(int p = 0; p<4; p++) ///ToTal MEMBER 1 Night Meal {

for(int q = 0; q<1; q++) {

for(int r = 2; r<3; r++)

{

Night1 += day[p][q][r]; } }

for(int p = 0; p<4; p++) ///ToTal MEMBER 2 Night Meal {

for(int q = 1; q<2; q++) {

for(int r = 2; r<3; r++) {

Night2 += day[p][q][r]; } }

int Night[2] = {Night1,Night2};

for(int p = 0; p<4; p++) ///ToTal MEMBER 1 Meal {

for(int q = 0; q<1; q++) {

for(int r = 0; r<3; r++) {

TotalMember1Meal += day[p][q][r]; } }

for(int p = 0; p<4; p++) ///ToTal MEMBER 2 Meal {

for(int q = 1; q<2; q++) {

for(int r = 0; r<3; r++) {

TotalMember2Meal += day[p][q][r]; } }

int TotalMemberMeal[2] = {TotalMember1Meal,TotalMember2Meal};

double Cost[4] = {180,305.50,125.50,90}; ///Total Shopping

double TotalCost = 0.0;

for(int i = 0; i<4; i++)

TotalCost += Cost[i];

double mealrate=Meal\_Rate(TotalMeal,TotalCost); view\_Print(mess,owner,member,Morning,Noon,Night,TotalMemberMeal,mealrate,Cost,TotalCost,TotalMorningMeal,TotalNoonMeal,TotalNightMeal,TotalMeal,date);

FILE \*fp;

char \*filename;

filename="mealinfo.csv";

fp=fopen(filename,"w+");

fprintf(fp,"Member,Day No.,Date,Morning,Noon,Night,Total\_Meal");

fprintf(fp,"\n\ntipu");

int sum=0;

for(int i=0; i<4; i++) {

sum=0;

fprintf(fp,",Day %d.,%d/12/18,",i+1,i+1);

for(int k=0; k<1; k++) {

for(int j=0; j<3; j++) {

fprintf(fp,"%d,",day[i][0][j]);

sum+=day[i][0][j]; }

fprintf(fp,"%d\n",sum); } }

fprintf(fp,",TOTAL,,%d,%d,%d,%d\n",Morning1,Noon1,Night1,TotalMember1Meal);

fprintf(fp,"\n,Meal Rate,%lf,,Deposit,%lf,,Cost,%lf,,Balance,%lf\n\n\n",mealrate,2000.00,(mealrate\*TotalMember1Meal),2000.00-(mealrate\*TotalMember1Meal));

fprintf(fp,"Member,Day No.,Date,Morning,Noon,Night,Total\_Meal\n");

fprintf(fp,"\nRobin");

int sum2=0;

for(int i=0; i<4; i++) {

sum2=0;

fprintf(fp,",Day %d.,%d/12/17,",i+1,i+1);

for(int k=1; k<2; k++) {

for(int j=0; j<3; j++) {

fprintf(fp,"%d,",day[i][1][j]);

sum2+=day[i][1][j]; }

fprintf(fp,"%d\n",sum2); } }

fprintf(fp,",TOTAL,,%d,%d,%d,%d\n",Morning1,Noon1,Night1,TotalMember1Meal);

fprintf(fp,"\n,Meal Rate,%lf,,Deposit,%lf,,Cost,%lf,,Balance,%lf\n",mealrate,1500.00,(mealrate\*TotalMember2Meal),1500.00-(mealrate\*TotalMember2Meal));

return 0; }

Sample Output:

Mess Name : SARKER VILLA

Mess Owner's Name : SARKER

Date of Birth : 04/01/1955

-----------------------------------------------------

-----------------------------------------------------

Member 1 :

NAME : Abid

DATE OF BIRTH : 10/01/1995

---------------------------------------

Deposit : 1500

Total Morning Meal : 3

Total Noon Meal : 3

Total Night Meal : 4

Total Meal : 10

Balance : 1149.5

Member 2 :

NAME : Adnan

DATE OF BIRTH : 15/2/1996

---------------------------------------

Deposit : 1400

Total Morning Meal : 4

Total Noon Meal : 4

Total Night Meal : 2

Total Meal : 10

Balance : 1049.5

\_\_\_\_\_\_\_\_\_MEAL INFORMATION\_\_\_\_\_\_\_\_\_

Total Deposit : 2900

Daily Shopping Costs-->

Day 1-09/10/2018 : 180

Day 2-09/10/2018 : 305.5

Day 3-24/10/2018 : 125.5

Day 4-24/10/2019 : 90

Total Cost : 701

Meal Rate : 35.05

Total Morning Meal : 7

Total Noon Meal : 7

Total Night Meal : 6

Total Meal : 20

Total Balance : 2199

Process returned 0 (0x0) execution time : 0.047 s

Press any key to continue.

Problem No: 3

Problem Name: CSE department wants to select Programming Coach for its students. Any student of the department can be a coach. He need to have high profile at least three ACM regional contest participation and number of problem solutions of ACM need to more 300. Students need to apply in the department, if anyone’s performance is below the requirement he will discard automatically. As a student of CSE, write OOP code for the project. All the communications will be held by message.

Source Code:

#include<bits/stdc++.h>

using namespace std;

class Date

{

public:

int day,month,year;

Date() {};

Date(int d,int m,int y)

{

day=d;

month=m;

year=y;

}

};

class Person ///Person class

{

public:

string name;

Date date;

Person() {}; ///Constructor-1 /Default Constructor

Person(string name)

{

this->name=name;

}

Person(string n,Date date)

{

name=n;

this->date=date;

}

};

class Department

{

public:

string dept\_name;

Department() {};

Department(string n)

{

dept\_name=n;

}

};

class Student : public Person /// Student class

{

public:

int id,ACM,prob\_solve;

Department department;

Student() {}; ///Constructor-1 /Default Constructor

Student(string name,Date \*date,Department \*department,int dd,int acm,int solve) : Person(name,\*date) ///constructor-2

{

id=dd;

this->department=\*department;

ACM=acm;

prob\_solve=solve;

}

View\_Student\_Info() ///function for viewing student info

{

cout<<"Student name: "<<name<<endl;

cout<<"Department: "<<department.dept\_name<<endl;

cout<<"Students Birthday: "<<this->date.day<<"-"<<this->date.month<<"-"<<this->date.year<<endl;

}

};

class Faculty : public Person ///faculty class

{

public:

Student student;

string designation;

Date date;

Department department;

Faculty() {}; ///Constructor-1 /Default Constructor

Faculty(string name,string designation,Date \*date,Department \*department):Person(name,\*date) ///constructor-2

{

this->designation=designation;

this->department=\*department;

}

View\_Faculty() ///faculty viewing function

{

cout<<"Name: "<<name<<endl;

cout<<"Department: "<<department.dept\_name<<endl;

cout<<"Designation: "<<designation<<endl;

}

int select\_coach(Student \*student)

{

if(student->ACM>=3&&student->prob\_solve>300)

return 1;

else

return 0;

}

};

int main()

{

Department \*department[5]= ///department type object declaration

{

new Department("ICT"),

new Department("CSE"),

new Department("CSE"),

new Department("ICT"),

new Department("CSE"),

};

Date \*date[6]= ///date type object declaration

{

new Date(22,02,1998),

new Date(28,06,1997),

new Date(0,05,1996),

new Date(25,01,1999),

new Date(31,10,1982),

new Date(12,12,1989)

};

Student \*student[3]= ///Student type object declaration

{

new Student("Sami",date[0],department[0],2001,4,201),

new Student("Tarik",date[1],department[1],2002,4,350),

new Student("Rafiq",date[2],department[2],2003,2,300)

};

Faculty \*faculty[2]= ///faculty object declaration.

{

new Faculty("Kamal Hossain Chowdhury","Assistant Professor",date[1],department[2]),

new Faculty("Mahmudul Hasan","Assistant Professor",date[2],department[4])

};

cout<<"THE DETAILS OF FACULTY MEMBER: "<<endl<<endl;

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

faculty[i]->View\_Faculty();

cout<<endl;

}

cout<<endl<<endl;

cout<<"THE DETAILS OF STUDENTS: "<<endl<<endl;

for(int i=0; i<3; i++){

student[i]->View\_Student\_Info();

cout<<endl;

}

int x;

cout<<endl<<endl;

for(int i=0;i<3;i++)

{

x=faculty[i]->select\_coach(student[i]);

if(x==1)

{

cout<<"SELECTED AS COACH:"<<endl;

cout<<"THE DETAILS OF THE STUDENT IS: "<<endl;

student[i]->View\_Student\_Info();

cout<<endl<<endl;

}

else

{

cout<<"NOT SELECTED AS COACH:"<<endl;

cout<<"THE DETAILS OF THE STUDENT IS: "<<endl;

student[i]->View\_Student\_Info();

cout<<endl<<endl;

}

}

return 0;

}

Sample Output:

THE DETAILS OF FACULTY MEMBER:

Name: Kamal Hossain Chowdhury

Department: CSE

Designation: Assistant Professor

Name: Mahmudul Hasan

Department: CSE

Designation: Assistant Professor

THE DETAILS OF STUDENTS:

Student name: Sami

Department: ICT

Students Birthday: 22-2-1998

Student name: Tarik

Department: CSE

Students Birthday: 28-6-1997

Student name: Rafiq

Department: CSE

Students Birthday: 0-5-1996

NOT SELECTED AS COACH:

THE DETAILS OF THE STUDENT IS:

Student name: Sami

Department: ICT

Students Birthday: 22-2-1998

SELECTED AS COACH:

THE DETAILS OF THE STUDENT IS:

Student name: Tarik

Department: CSE

Students Birthday: 28-6-1997

NOT SELECTED AS COACH:

THE DETAILS OF THE STUDENT IS:

Student name: Rafiq

Department: CSE

Students Birthday: 0-5-1996

Process returned 0 (0x0) execution time : 0.047 s

Press any key to continue.