Q1: 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("MASTER VILLA");

Owner owner("Shuvo","09/11/1998");

Member \*member[2]= {

new Member("jwel","01/2/1998",2000),

new Member("navin","2/12/1999",1900), };

Date \*date[4]= {

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

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

new Date("11/10/2018"),

new Date("12/10/2019"), };

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

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

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

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] = {210.50,105.75,225.50,90.75}; ///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]; }

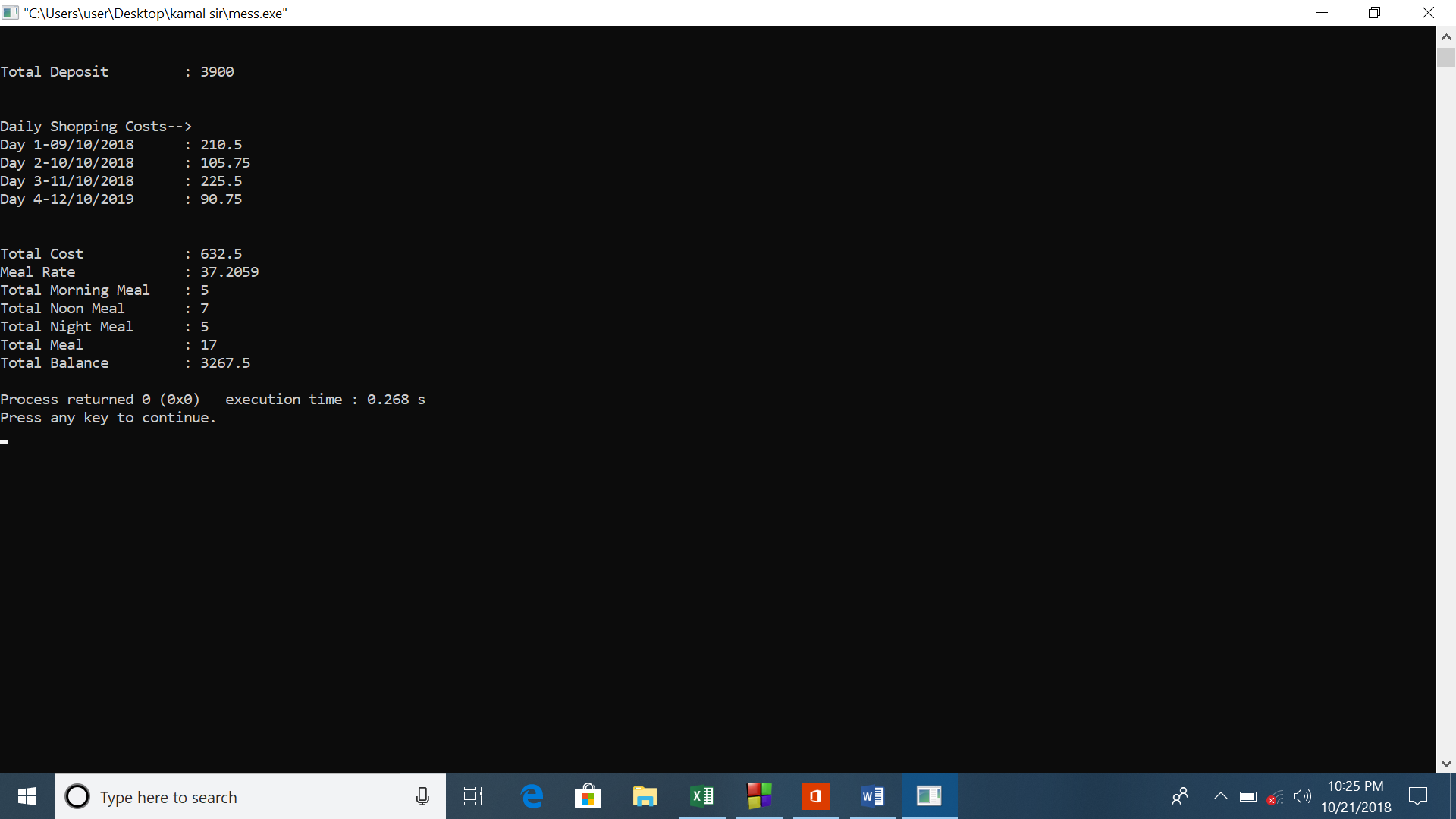
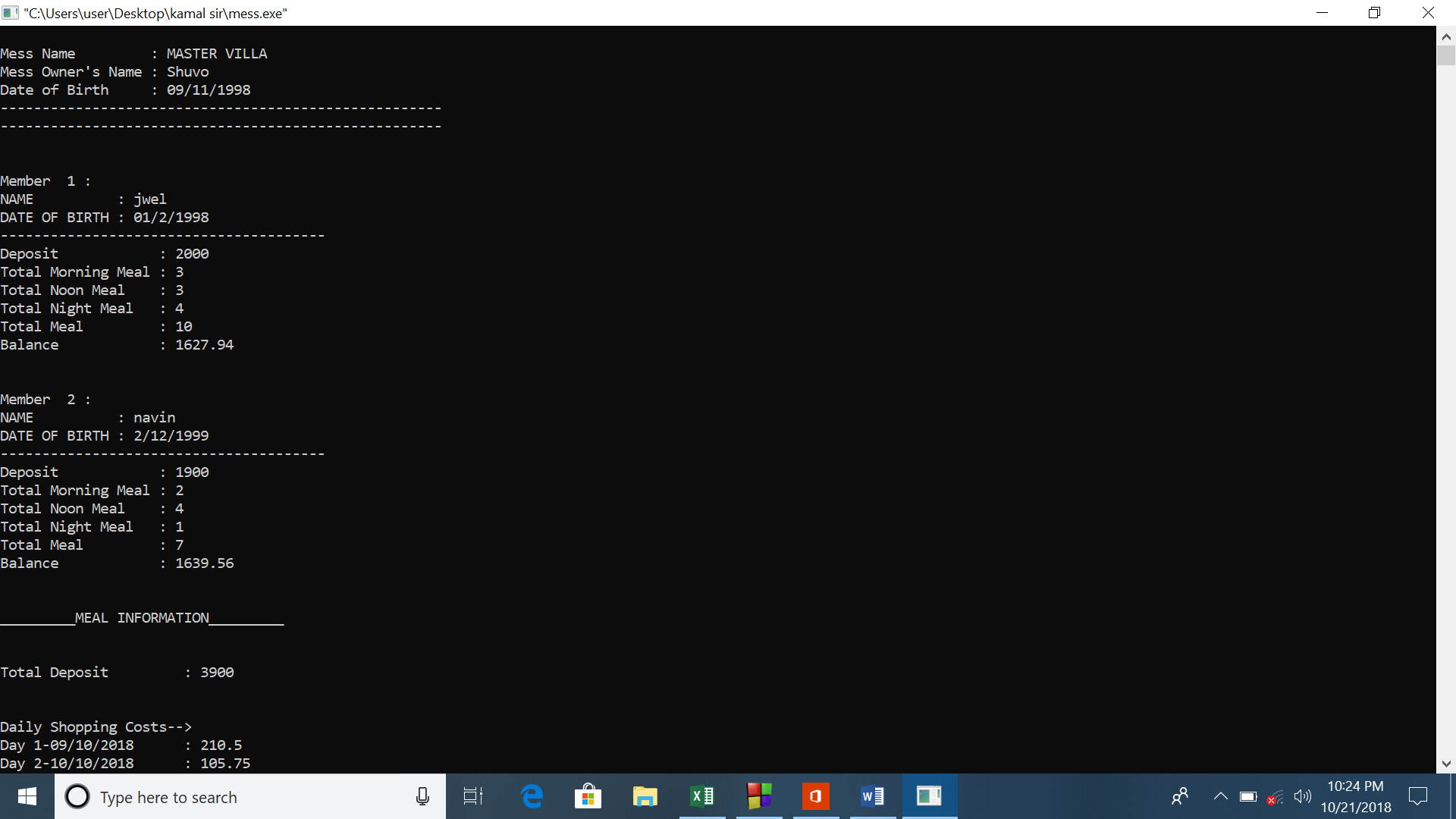
fprintf(fp,"%d\n",sum2); } }

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:



Q2: 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 preordered 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("hasan","20-6-1996",department,"1001"),

new Student("jisan","13-3-1998",department,"1002"),

new Student("sobuj","21-11-1996",department,"1003"),

new Student("sajib","05-07-1997",department,"1004")

};

Sellsman sells\_person("Saiful Islam","13-03-1985",1);

sells\_person.view\_sellsman();

Login \*login\_data[2]={

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

new Login("KAL","222")

};

Faculty \*faculty\_member[2]=

{

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

new Faculty("Lecturer","Rabiul 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;

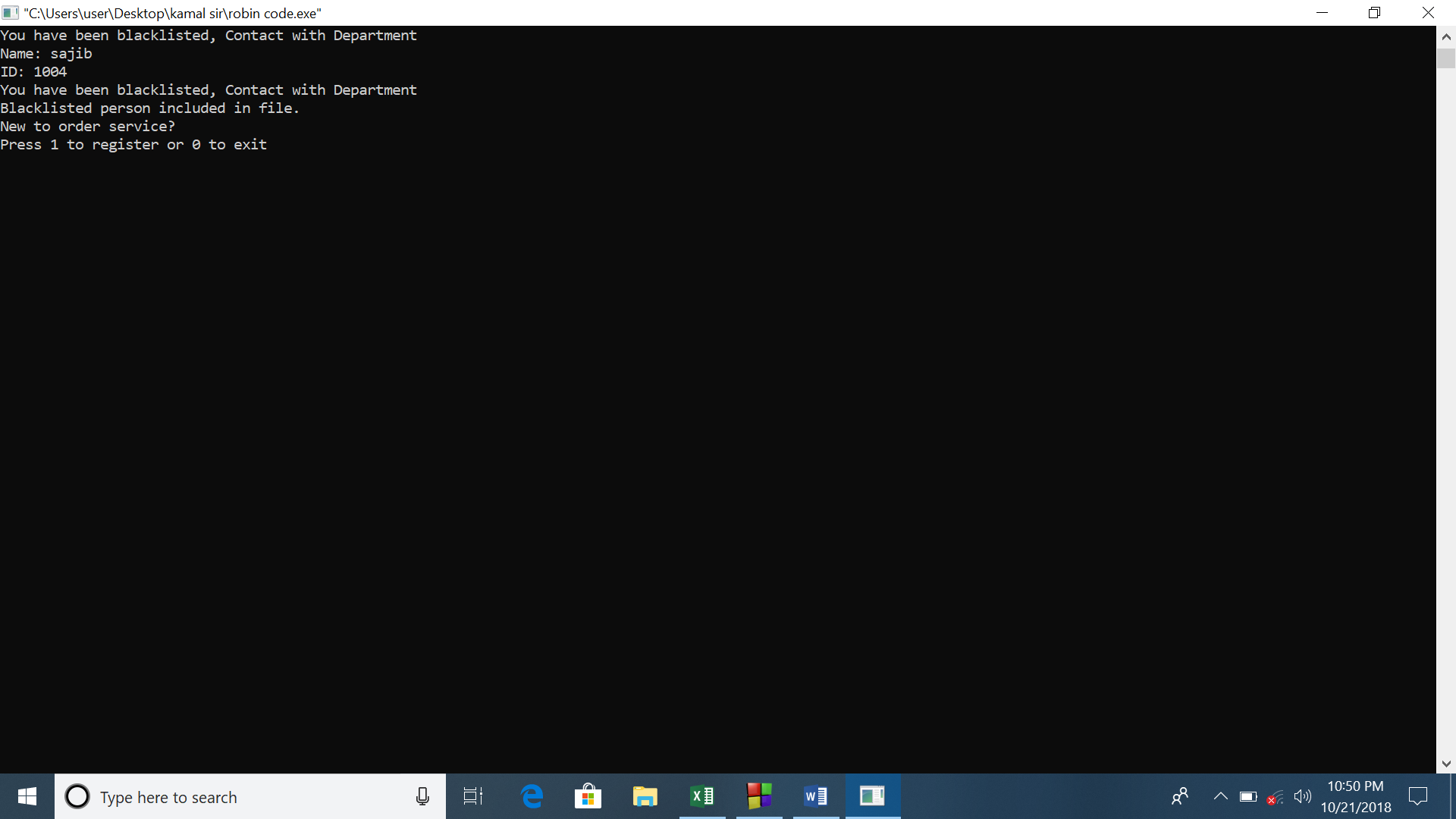
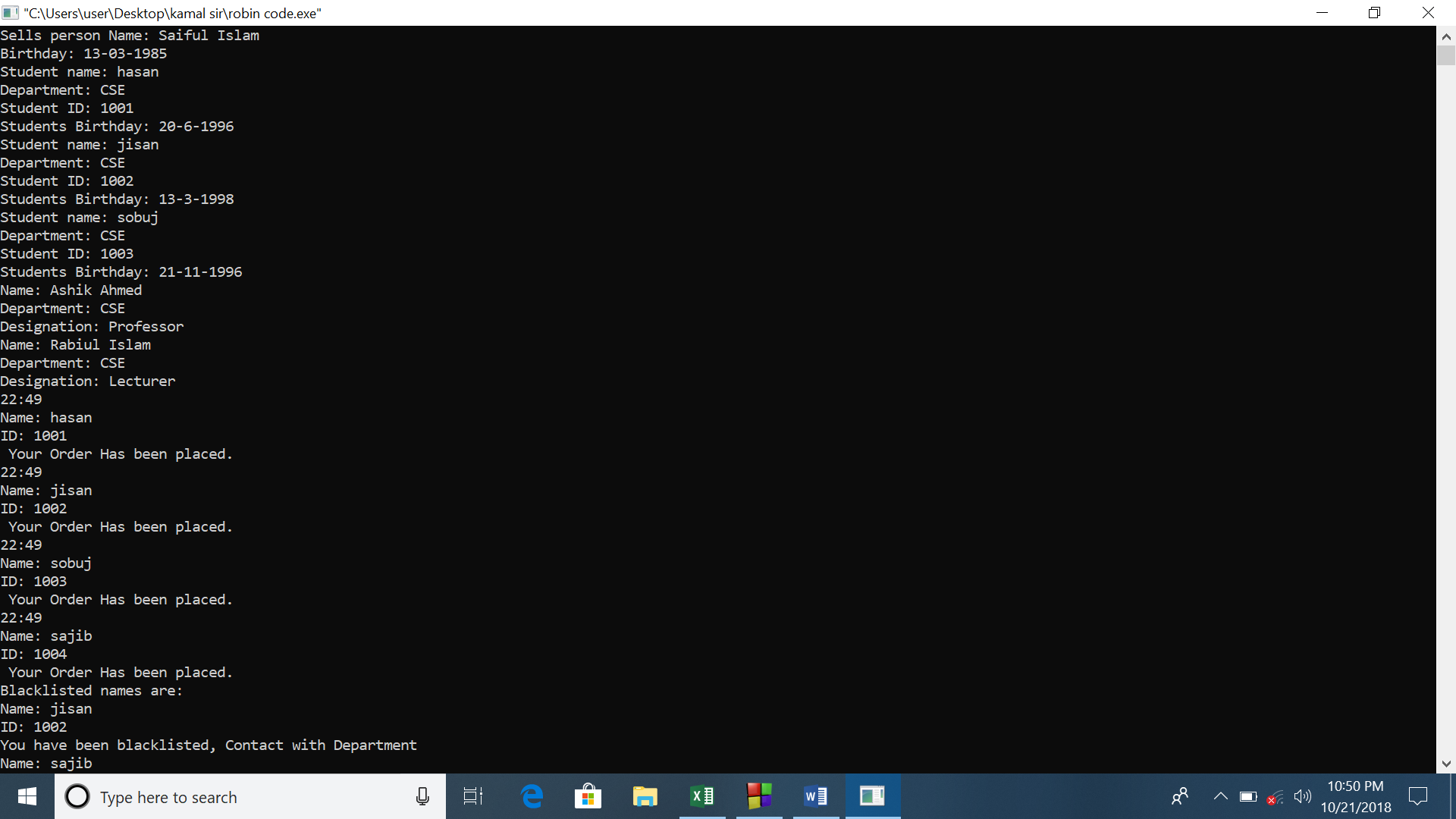
else

orderfood(students[i]);

}

}

SAMPLE OUTPUT:



Q3: Write C++ code in object oriented approach for the students. Student can be rewarded from the department if he gets good GPA in a semester and solved atleast 100 ACM problems in the last year. Department will publish the top 5 students name in their honor board. As a student of CSE, write OOP code for the project.

Source code:

#include<bits/stdc++.h>

using namespace std;

float arr[9];

int i,cnt=0;

class Person

{

protected:

string name;

public:

Person() { }

Person(string name)

{

this->name=name;

}

string acces\_person()

{

return name;

}

};

class Contestant

{

protected:

int ACM\_solve;

};

class Student : public Contestant,public Person

{

protected:

float GPA;

// string name;

public:

Student() { }

Student(string Name,float GPA,int ACM\_solve):Person(Name)

{

this->GPA=GPA;

this->ACM\_solve=ACM\_solve;

//name=N;

}

int checkgpa(float ab)

{

if(GPA==ab and ACM\_solve>=100)

return 1;

else return 2;

}

float accessGPA()

{

return GPA;

}

void show\_student\_info()

{

cout << setw(10) << acces\_person() << setw(13) << GPA << endl << endl;

}

};

class Teacher : public Person

{

protected:

string designation,Username,Password;

public :

Teacher() { }

Teacher (string name,string designation,string Username, string Password):Person(name)

{

this->designation=designation;

this->Username=Username;

this->Password=Password;

}

string access\_username(){ return Username;}

string access\_password(){ return Password;}

void show\_teacher\_info()

{

cout << endl

<<"This List has been Prepared by" << endl

<< acces\_person() << endl << designation << endl;

}

int Match\_info(string Username,string Password)

{

if(this->Username==Username and this->Password==Password)

return 1;

else return 2;

}

void Bubblesort()

{

int i,j;

float temp;

for(i=0;i<8-1;i++)

for(j=0;j<8-i-1;j++)

{

if(arr[j]>arr[j+1])

{

temp=arr[j+1];

arr[j+1]=arr[j];

arr[j] = temp ;

}

}

}

};

class Department

{

protected:

string Department\_name;

string saved\_username[3],saved\_password[3];

public:

Department() { Department\_name="Department of Computer Science and Engineering"; }

void Save\_username\_password(string j,string k,int l)

{

saved\_username[l]=j;

saved\_password[l]=k;

}

int Access\_Request1(string Username)

{

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

{

if(saved\_username[i]==Username)

{

cnt=1;

return i;

}

}

}

int Access\_Request2(string Password,int Index)

{

if(Password==saved\_password[Index])

return 1;

}

void show\_dept\_name()

{

cout << Department\_name << endl << endl;

}

};

class Date

{

public :

string Month[14];

void Month\_initializer()

{

Month[0]=" ",Month[1]="January",Month[2]="February",Month[3]="March",Month[4]="April",Month[5]="May",Month[6]="June",Month[7]="July",Month[8]="August",Month[9]="September",Month[10]="October",Month[11]="November",Month[12]="December";

}

int year,month,date,x;

void Incertpresenttime(tm \*ltm)

{

year=ltm->tm\_year + 1900 ;

month=ltm->tm\_mon + 1;

date=ltm->tm\_mday ;

}

void display()

{

cout << "Date of publication " << endl;

cout << date << "th " << Month[month] << " ," << year << endl;

}

};

int main()

{

int n,a,b,j,Index=-1;

Student \*performance[9]={

new Student("ashik",3.07 , 200 ),

new Student("Sweet",3.31,101),

new Student("mahedi",3.12,45),

new Student("forhad",3.36,34),

new Student("tuhin",3.01,199),

new Student("Ashik",3.76,300),

new Student("mamun",3.26,267),

new Student("shakil",3.16,198)

};

Department CSE;

Teacher \*assigned\_teacher[3]={

new Teacher("Kamal Hossain Chowdhury", "Chairman","Kamal\_Hossain","kamal1234" ),

new Teacher("Mahmudul Hasan Raju", "Assistant professor", "Mahmudul\_Hasan","raju4321"),

new Teacher("Faisal Bin Abdul Aziz", "Assistant professor","Faisal\_Aziz","faisal1234")

};

string temp\_username,temp\_password;

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

{

temp\_username=assigned\_teacher[i]->access\_username();

temp\_password=assigned\_teacher[i]->access\_password();

CSE.Save\_username\_password(temp\_username,temp\_password,i);

}

string Username,Password;

cout << "Enter the Username" << endl;

cin >> Username;

Index=CSE.Access\_Request1(Username);

while(cnt!=1)

{

cout << "Invalid Username. Enter again" << endl;

cin >> Username;

Index=CSE.Access\_Request1(Username);

}

cnt=0;

cout << "Enter the Password" << endl;

cin >> Password;

cnt=CSE.Access\_Request2(Password,Index);

while(cnt!=1)

{

cout << "wrong password. Enter again" << endl;

cin >> Password;

cnt=CSE.Access\_Request2(Password,Index);

}

float temp;

int counter=0;

float maximum;

for(i=0;i<8;i++)

{

arr[i]=performance[i]->accessGPA();

}

assigned\_teacher[Index]->Bubblesort();

cout << " TOP 5 students on the honor Board are" << endl << endl;

cout << setw(10) << "Student Name " << setw(10) << "GPA" << endl ;

cout << setw(10) << "\_\_\_\_\_\_\_\_\_\_\_\_\_" << setw(10) << "\_\_\_\_" << endl << endl ;

for(i=7;i>=0;i--)

{

maximum=arr[i];

for(j=0;j<8;j++)

{

if(performance[j]->checkgpa(maximum)==1)

{

counter++;

performance[j]->show\_student\_info();

}

if(counter==5) break;

}

}

assigned\_teacher[Index]->show\_teacher\_info();

CSE.show\_dept\_name();

time\_t now = time(0);

tm \*ltm = localtime(&now);

Date A;

A.Incertpresenttime(ltm);

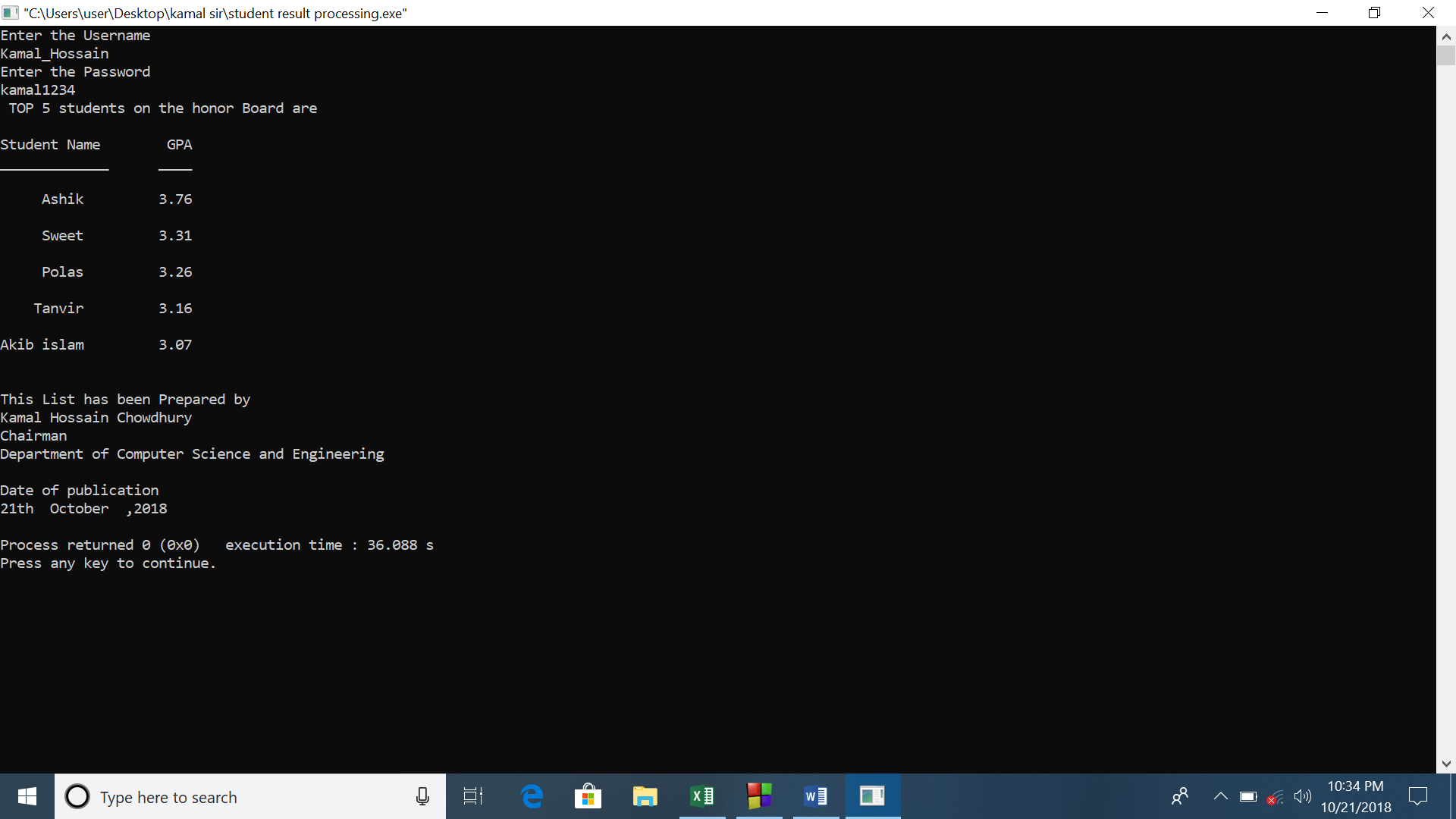
A.Month\_initializer();

A.display();

return 0;

}

SAMPLE OUTPUT:



Q4: 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("CSE"),

new Department("CSE"),

new Department("CSE"),

new Department("CSE"),

new Department("CSE"),

};

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

{

new Date(23,04,1997),

new Date(13,05,1998),

new Date(01,03,1996),

new Date(12,05,1997),

new Date(19,04,1980),

new Date(22,07,1981)

};

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

{

new Student("Sejan",date[0],department[0],1001,4,301),

new Student("Polash",date[1],department[1],1002,4,400),

new Student("Shanto",date[2],department[2],1003,2,200)

};

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

{

new Faculty("Kamal Hossain Chowdhury","Assistant Professor",date[3],department[3]),

new Faculty("Mahmudul Hasan","Assistant Professor",date[4],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:

