**Problem no 01:-** 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.

**Input:**

#include<iostream>

#include<stdio.h>

#include<conio.h>

#include<fstream>

#include<string.h>

using namespace std;

class MealBazzar

{

public:

int Meal, Cost, Total\_meal, Total\_cost;

int Meal1, Cost1, permeal;

MealBazzar()

{

int couns = 0;

cout << "enter todays meal: ";

cin >> Meal;

cout << "enter todays cost: ";

cin >> Cost;

ofstream fout("Meal.txt", ios :: app);

fout << Meal << " " << Cost << endl;

fout.close();

couns++;

cout << "you entered meal and cost" << couns << " times" << endl;

}

int display()

{

Total\_meal = 0;

Total\_cost = 0;

ifstream fin("Meal.txt", ios :: app);

while(!fin.eof())

{

fin >> Meal1;

fin >> Cost1;

Total\_meal = Total\_meal + Meal1;

Total\_cost = Total\_cost + Cost1;

Meal1 = 0;

Cost1 = 0;

}

cout << "Total meal : " << Total\_meal << endl;

cout << "Total cost: " << Total\_cost << endl;

cout << "Cost per meal: " << (double)Total\_cost / Total\_meal << endl;

}

};

class MealBazzar1

{

public:

int Meal, Cost, Total\_meal, Total\_cost;

int Meal1, Cost1, permeal;

MealBazzar1()

{

int couns = 0;

cout << "enter todays meal: ";

cin >> Meal;

cout << "enter todays cost: ";

cin >> Cost;

ofstream fout("one.txt", ios :: app);

fout << Meal<< " " << Cost << endl;

fout.close();

couns++;

cout << "you enter meal and cost" << couns << " times" << endl;

}

int dispalay()

{

Total\_meal = 0;

Total\_cost = 0;

ifstream fin("one.txt", ios :: app);

while(!fin.eof())

{

fin >> Meal1;

fin >> Cost1;

Total\_meal = Total\_meal + Meal1;

Total\_cost = Total\_cost + Cost1;

Meal1 = 0;

Cost1 = 0;

}

cout << "Total meal: " << Total\_meal << endl;

cout << "Total cost: " << Total\_cost << endl;

cout << "Cost per meal: " << (double)Total\_cost / Total\_meal << endl;

}

};

class Habib

{

public:

int input()

{

char name[100];

x:

cout << "enter name: " << endl;

cin >> name;

if(strcmp(name, "Habib") == 0)

{

cout << "his / her name: " << name << endl;

cout << "his / her occupation: student " << endl;

cout << "his / her phone number: 015\*\*\*\*\*\*\*\*" << endl;

cout << "his / her district: comilla" << endl;

}

else

{

cout << "you type wrong name" << endl;

goto x;

}

}

};

class Ahsan

{

public:

int input()

{

char name[100];

y:

cout << "enter name: " << endl;

cin >> name;

if(strcmp(name, "Ahsan") == 0)

{

cout << "his / her name: " << name << endl;

cout << "his / her occupation: student " << endl;

cout << "his / her phone number: 015\*\*\*\*\*\*\*\*" << endl;

cout << "his / her district: comilla" << endl;

}

else

{

cout << "you type wrong name" << endl;

goto y;

}

}

};

int main()

{

int x;

cout << "\n\n\t\t\t MESS MANAGEMENT SYSTEM";

cout << "\n\t\t\t ......................";

do

{

cout << "\n\n enter your choise:- ";

cout << "\n\n 1. Habib";

cout << "\n\n 2. Ahsan";

cout << "\n\n enter your choise:- ";

cin >> x;

switch(x)

{

case 1:

{

Habib habib;

habib.input();

MealBazzar obj;

obj.display();

break;

}

case 2:

{

Ahsan ahsan;

ahsan.input();

MealBazzar obj;

obj.display();

break;

}

default:

{

cout << "you press the wrong key\n\n";

break;

}

}

}

while(1);

}

**Output:**

MESS MANAGEMENT SYSTEM

.............................................

enter your choise:-

1. Habib

2. Ahsan

enter your choise:- 1

enter name:

Habib

his / her name: Habib

his / her occupation: student

his / her phone number: 015\*\*\*\*\*\*\*\*

his / her district: comilla

enter todays meal: 3

enter todays cost: 150

you entered meal and cost1 times

Total meal : 13

Total cost: 670

Cost per meal: 51.5385

**Problem no 02:-**  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.

**Input:**

#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 date1;

Person(){};

Person(string name)

{

this->name=name;

}

Person(string name,string date)

{

this->name=name;

date1.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: "<<date1.date<<endl;

}

friend orderfood();

friend blacklist();

};

class Sellsman : public Person

{

public:

int seller\_no;

Sellsman(){};

Sellsman(string name,string date2,int seller\_no): Person(name,date2)

{

this->seller\_no=seller\_no;

};

view\_sellsman()

{

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

cout<<"Birthday: "<<date1.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("Habib","24-03-1997",department,"1011"),

new Student("Shipon","1-6-1998",department,"1022"),

new Student("Anas","30-09-1996",department,"1033"),

new Student("Manajir","04-10-1998",department,"1044")

};

Sellsman sells\_person("Afzal Hossain","1-05-1965",1);

sells\_person.view\_sellsman();

Login \*login\_data[2]={

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

new Login("KAL","222")

};

Faculty \*faculty\_member[2]=

{

new Faculty("Professor","Kamal Hossain Chawdhory",department,login\_data[0]),

new Faculty("Lecturer","Faysal Ahmed",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\_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"<<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]);

}

}

**Output:**

Sells person Name: Afzal Hossain

Birthday: 1-05-1965

Student name: Habib

Department: CSE

Student ID: 1011

Students Birthday: 24-03-1997

Student name: Shipon

Department: CSE

Student ID: 1022

Students Birthday: 1-6-1998

Student name: Anas

Department: CSE

Student ID: 1033

Students Birthday: 30-09-1996

Name: Kamal Hossain Chawdhory

Department: CSE

Designation: Professor

Name: Faysal Ahmed

Department: CSE

Designation: Lecturer

1:26

Name: Habib

ID: 1011

Your Order Has been placed.

1:26

Name: Shipon

ID: 1022

Your Order Has been placed.

1:26

Name: Anas

ID: 1033

Your Order Has been placed.

1:26

Name: Manajir

ID: 1044

Your Order Has been placed.

Blacklisted names are:

Name: Shipon

ID: 1022

You have been blacklisted, Contact with Department

Name: Manajir

ID: 1044

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 03:-** 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.

**Input:**

#include<bits/stdc++.h>

using namespace std;

class Person

{

public:

string name,NID,DOB;

void getInfo(string x,string y,string z)

{

name=x;

NID=y;

DOB=z;

}

void putInfo()

{

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

}

};

class Student:public Person

{

public:

int ID,solved;

double gpa;

void getData(string x,string y,string z,int a,int b,double c)

{

getInfo(x,y,z);

ID=a;

solved=b;

gpa=c;

}

void putData()

{

putInfo();

cout<<"ID of the student : "<<ID<<endl;

}

bool qualify()

{

if(solved>=100 && gpa>=3.5)

return true;

else

return false;

}

bool operator < (Student A)

{

if((((gpa\*10)/4.0)+((solved\*10)/100.0)) > (((A.gpa\*10)/4.0)+((A.solved\*10)/100.0)))

return true;

return false;

}

};

int main()

{

int i,j,n,cnt=0;

cout<<"Enter the number of student : ";

cin>>n;

Student obj[n];

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

{

string x,y,z;

int a,b;

double c;

cout<<"Give Data for Student : "<<i+1<<endl;

cout<<"Enter the name,NID & Date of birth of the student: \n";

cin>>x>>y>>z;

cout<<"Enter the ID, Solved problem & GPA of the student: \n";

cin>>a>>b>>c;

obj[i].getData(x,y,z,a,b,c);

}

sort(obj,obj+n);

j=0;

while(1)

{

if(cnt>4 || j>=n)

break;

if(obj[j].qualify())

{

cout<<"Rank : "<<cnt+1<<endl;

obj[j].putData();

cout<<endl;

cnt+=1;

}

j++;

}

return 0;

}

**Output:**

Enter the number of student : 8

Give Data for Student : 1

Enter the name,NID & Date of birth of the student:

Anas 9238778 23409875

Enter the ID, Solved problem & GPA of the student:

25 150 3.5

Give Data for Student : 2

Enter the name,NID & Date of birth of the student:

Manajir 14984 409374

Enter the ID, Solved problem & GPA of the student:

30 120 3.6

Give Data for Student : 3

Enter the name,NID & Date of birth of the student:

Boby 148743187 41803478

Enter the ID, Solved problem & GPA of the student:

35 500 3.55

Give Data for Student : 4

Enter the name,NID & Date of birth of the student:

Alkas 24389745 3420947

Enter the ID, Solved problem & GPA of the student:

21 111 3.6

Give Data for Student : 5

Enter the name,NID & Date of birth of the student:

Saiful 18734873 4130987489

Enter the ID, Solved problem & GPA of the student:

24 160 3.5

Give Data for Student : 6

Enter the name,NID & Date of birth of the student:

Rakibul 1398474 34180748

Enter the ID, Solved problem & GPA of the student:

56 260 3.2

Give Data for Student : 7

Enter the name,NID & Date of birth of the student:

Reza 2389748 342874

Enter the ID, Solved problem & GPA of the student:

29 1022 3.5

Give Data for Student : 8

Enter the name,NID & Date of birth of the student:

Mushfique 10987438 140934

Enter the ID, Solved problem & GPA of the student:

56 111 4.0

Rank : 1

Name : Reza

ID of the student : 29

Rank : 2

Name : Boby

ID of the student : 35

Rank : 3

Name : Anas

ID of the student : 25

Rank : 4

Name : Saiful

ID of the student : 24

Rank : 5

Name : Manajir

ID of the student : 30

**Problem No-04:-** . 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.

**Input:**

#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

{

public:

string name;

Date d1;

Person() {};

Person(string n)

{

name=n;

}

Person(string n,Date \*d)

{

name=n;

d1=\*d;

}

};

class Department

{

public:

string dept\_name;

Department() {};

Department(string n)

{

dept\_name=n;

}

};

class Student : public Person

{

public:

int id,ACM,prob\_solve;

Department d;

Student() {};

Student(string n,Date \*d2,Department \*dp,int dd,int acm,int solve) : Person(n,d2)

{

id=dd;

d=\*dp;

ACM=acm;

prob\_solve=solve;

}

View\_Student\_Info()

{

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

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

cout<<"Students Birthday: "<<d1.day<<"-"<<d1.month<<"-"<<d1.year<<endl;

cout<<"NUMBER OF ACM PARTICIPATION: "<<ACM<<endl;

cout<<"NUMBER OF SOLVED PROBLEMS: "<<prob\_solve<<endl;

}

};

class Faculty : public Person

{

public:

Student \*s;

string designation;

Department dp;

Faculty() {};

Faculty(string d,string n,Department \*dp1):Person(d)

{

designation=n;

dp=\*dp1;

}

View\_Faculty()

{

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

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

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

}

int select\_coach(Student \*s1)

{

s=s1;

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

return 1;

else

return 0;

}

};

int main()

{

Department \*dp[5]=

{

new Department("CSE"),

new Department("CSE"),

new Department("CSE"),

new Department("CSE"),

new Department("CSE"),

};

Date \*dt[4]=

{

new Date(10,01,1994),

new Date(12,03,1980),

new Date(14,05,1992),

new Date(16,07,1985)

};

Student \*s[3]=

{

new Student("Jafour",dt[0],dp[0],1011,4,300),

new Student("Kamal",dt[1],dp[1],1022,4,350),

new Student("Abzar",dt[2],dp[2],1033,2,250)

};

Faculty \*f[2]=

{

new Faculty("Helal","Professor",dp[3]),

new Faculty("Ainan","Lecturer",dp[4])

};

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

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

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

cout<<endl;

}

cout<<endl<<endl;

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

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

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

cout<<endl;

}

int x;

cout<<endl<<endl;

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

{

x=f[i]->select\_coach(s[i]);

if(x==1)

{

cout<<"THE STUDENT CAN BE SELECTED AS COACH.."<<endl;

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

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

cout<<endl<<endl;

}

else

{

cout<<"THE STUDENT IS NOT SELECTED AS COACH.."<<endl;

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

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

cout<<endl<<endl;

}

}

return 0;

}

**Output:**

THE DETAILS OF FACULTY MEMBER:

Name: Helal

Department: CSE

Designation: Professor

Name: Ainan

Department: CSE

Designation: Lecturer

THE DETAILS OF STUDENTS:

Student name: Jafour

Department: CSE

Students Birthday: 10-1-1994

NUMBER OF ACM PARTICIPATION: 4

NUMBER OF SOLVED PROBLEMS: 300

Student name: Kamal

Department: CSE

Students Birthday: 12-3-1980

NUMBER OF ACM PARTICIPATION: 4

NUMBER OF SOLVED PROBLEMS: 350

Student name: Abzar

Department: CSE

Students Birthday: 14-5-1992

NUMBER OF ACM PARTICIPATION: 2

NUMBER OF SOLVED PROBLEMS: 250

THE STUDENT IS NOT SELECTED AS COACH..

THE DETAILS OF THE STUDENT IS:

Student name: Jafour

Department: CSE

Students Birthday: 10-1-1994

NUMBER OF ACM PARTICIPATION: 4

NUMBER OF SOLVED PROBLEMS: 300

THE STUDENT CAN BE SELECTED AS COACH..

THE DETAILS OF THE STUDENT IS:

Student name: Kamal

Department: CSE

Students Birthday: 12-3-1980

NUMBER OF ACM PARTICIPATION: 4

NUMBER OF SOLVED PROBLEMS: 350

THE STUDENT IS NOT SELECTED AS COACH..

THE DETAILS OF THE STUDENT IS:

Student name: Abzar

Department: CSE

Students Birthday: 14-5-1992

NUMBER OF ACM PARTICIPATION: 2

NUMBER OF SOLVED PROBLEMS: 250

**Problem No-05:-** Like Uber, CNG owners want to develop a software which can communicate with customer and driver. A customer needs to say his/her destination and nearer CNG driver will get message from the Uber company with name and mobile number of the customer. The driver can make a direct call and if the customer confirms the ride then driver will come to spot and carry the passenger. As a student of CSE, write OOP code for the project. You need to pass message from one end to another end to make communications.

**Input:**

#include<iostream>

#include<math.h>

#include<fstream>

using namespace std;

class Person

{

protected:

string name;

int national\_id;

public:

Person(){}

Person(string name,int national\_id)

{

this->name=name;

this->national\_id=national\_id;

}

};

class Driver: public Person

{

int mobile\_number;

string current\_location;

public:

double lattitude;

double longitude;

Driver (){};

Driver(string name,int national\_id,int mobile\_number,string current\_location,double lattitude,double longitude):Person(name,national\_id)

{

this->mobile\_number=mobile\_number;

this-> current\_location=current\_location;

this-> lattitude=lattitude;

this-> longitude=longitude;

}

void driverfile ()

{

ofstream myfile2;

myfile2.open ("Driver Information.txt");

myfile2<<"\nDriver Details "<<endl;

myfile2<<"-------------------------------"<<endl;

myfile2<<"Mobile number :"<<mobile\_number<<endl;

myfile2<<"current location : "<<current\_location<<endl;

myfile2<<"Lattitude :"<<lattitude<<" degree "<<endl;

myfile2<<"Longitude : "<<longitude<<" degree "<<endl;

myfile2<<"Name :"<<name<<endl;

myfile2<<"National id :"<<national\_id<<endl;

myfile2<<"--------------------------------";

myfile2.close();

}

void booking ()

{

cout<<"a seat has been confirmed"<<endl;

}

};

class Uber

{

Driver driver1,driver2;

public:

void driver\_list(Driver &driver1,Driver &driver2)

{

this-> driver1=driver1;

this-> driver2=driver2;

}

double calculate\_distance( double customer\_lattitude, double customer\_longitude, double driver\_lattitude, double driver\_longitude )

{

int nRadius = 6371;

double nDLat = (driver\_lattitude - customer\_lattitude) \* (M\_PI/180);

double nDLon = (driver\_longitude-customer\_longitude) \* (M\_PI/180);

double nA = pow ( sin(nDLat/2), 2 ) + cos(customer\_lattitude) \* cos(driver\_lattitude) \* pow ( sin(nDLon/2), 2 );

double nC = 2 \* atan2( sqrt(nA), sqrt( 1 - nA ));

double distance = nRadius \* nC;

return distance;

}

void contact\_with\_driver(double customer\_lattitude,double customer\_longitude)

{

double distancefromdriver1,distancefromdriver2;

distancefromdriver1=calculate\_distance(customer\_lattitude,customer\_longitude,driver1.lattitude,driver1.longitude); distancefromdriver2=calculate\_distance(customer\_lattitude,customer\_longitude,driver2.lattitude,driver2.longitude);

if(distancefromdriver1<distancefromdriver2)

driver1.booking();

else

driver2.booking();

}

};

class Customer:public Person

{

int mobile\_number;

string destination;

double lattitude;

double longitude;

public:

Customer(){}

Customer(string name,int national\_id,int mobile\_number,string destination,double lattitude,double longitude):Person(name,national\_id)

{

this-> mobile\_number=mobile\_number;

this->destination=destination;

this-> lattitude=lattitude;

this-> longitude=longitude;

}

friend class Uber;

void customerfile ()

{

ofstream myfile1;

myfile1.open ("Customer Information.txt");

myfile1<<"\nCustomer Details "<<endl;

myfile1<<"----------------------------"<<endl;

myfile1<<"Mobile number :"<<mobile\_number<<endl;

myfile1<<"Lattitude :"<<lattitude<<" degree "<<endl;

myfile1<<"Longitude : "<<longitude<<" degree "<<endl;

myfile1<<"Destination : "<<destination<<endl;

myfile1<<"Name :"<<name<<endl;

myfile1<<"National id :"<<national\_id<<endl;

myfile1<<"----------------------------"<<endl;

}

void contact\_with\_uber(Uber &uber)

{

double lattitude,longitude;

this-> lattitude=lattitude;

this-> longitude=longitude;

cout<<"\n\nFor customer "<<this->name<<", whose Destination is "<<this->destination<<", ";

uber .contact\_with\_driver(lattitude,longitude);

}

};

int main()

{

Customer customer("Fazlul",1,2345,"Dhaka",23,91);

Driver driver1("Jamal",2,6789,"Comilla",21,93);

Driver driver2("Kashem",3,9876,"Chittagong",22,91);

driver1.driverfile();

customer.customerfile();

Uber uber;

uber.driver\_list(driver1,driver2);

customer.contact\_with\_uber(uber);

}

**Output:**

For customer Fazlul, whose Destination is Dhaka, a seat has been confirmed