24K-0559

BSCS2H

OOP THEORY ASSIGNMENT 2 :

QUESTION1).

CODE:

```
//24K-0559 BAZIL-UDDIN-KHAN
#include <iostream>
#include<cstring>
#include<string>
using namespace std;

class Person
{
   protected:
   string PersonName;
   string DateOfJoining;
   string email;
   string phoneNumber;

   public:
   Person(){}
```

```
    Person(string name,string date,string Email,string number) :
PersonName(name),DateOfJoining(date),email(Email),phoneNumber(
number)
    {}

    virtual  void DisplayInfo()
    {
        cout << " Person Details Are " << endl;
        cout << "Person Name is = " << PersonName << endl;
        cout << "Person Date Of Joining  is = " << DateOfJoining << endl;
        cout << "Person Email is = " << email << endl;
        cout << "Person Phone Number  is = " << phoneNumber << endl;

    }

    bool operator==(const Person& other) const
    {
    return this->email == other.email;
    }

};
```

```
class Routes
{
   private:
   string Route[24] =
{"Roùte2","Route3","Route3B","Route4","Route4B","Route5","Route
5B","Route6","Route6B","Route7","Route8","Route9","Route17","Ro
ute18","Route20","Route22","Route23","Route23B","Route24","Rout
e24B","Route25","Route25B","Route26","Route26B"};
   string RoutesPickUp[24] =
{"Saffora","SindhBaloach","KamranChorangi","Jamalipull","Maskan",
"PowerHouse","DolminMall","bardadari","Golbarq","Donisel","clifto
n","Bahria","Dha","UniversityRoad","Korangi","lyaripull","Motimehal
","Sheefaisal","Balochistansajji","Kdm","Mehran","Lalqila","Tariqroad
","Balochistanx"};

   public:
   int CheckRoute(string Routeu)
   {

      for(int i=0; i < 24; i++)
      {
         if(Route[i] == Routeu)
         {
```

```
            return 1;
        }
    }

  cout << " Not Found " << endl;
  return -1;

}

string GetRouteName(string RouteNumber)
{
    for(int i =0; i < 24;i++)
    {
        if(Route[i] == RouteNumber)
        {
            string val = RoutesPickUp[i];
            return(val);
        }
    }
    return " ";
}
```

```
void DisplayInfo()
{
   cout << " Routes Details Are " << endl;
   for( int y =0; y< 24;y++)
   {
      cout << Route[y] << "  ";
   }

   cout<< endl;

   cout << " Routes Stops Are " << endl;
    for(int y =0; y< 24;y++)
   {
      cout << RoutesPickUp[y] << "  ";
   }




}
bool operator==(const Routes& other) const
{
```

```
      for (int i = 0; i < 24; i++)
     {
       if (this->Route[i] != other.Route[i] || this->RoutesPickUp[i] != other.RoutesPickUp[i])
       {
          return false;  // Return false if any mismatch is found
        }
     }
      return true;
   }// Return true if all elements match




};

class PaymentFee
{
   private:
   double Fees;
   string CardStatus;
```

```
public:
PaymentFee()
{
   Fees =0;
   CardStatus = "NotPaid";
}
PaymentFee(double Fee)
{
   this->Fees = Fee;
}

int PaymentFees(double fees)
{
   if(Fees == fees)
   {
      this->Fees = Fees;
      this->CardStatus = "Paid";
      cout << " Payment Susssfully made " << endl;
      return 1;
   }
   else
   {
```

```
        cout << " Payment Not Recieved " << endl;
        return 0;
      }

  }

  void DisplayInfo()
  {
     cout << " Fees Status Is  " << endl;
     cout << " Fees Is " << Fees << endl;
     cout << " Card Status Is " << CardStatus << endl;
  }

  string GetStatus()
  {
     return CardStatus;
  }


};

class Student : public Person
```

```
{
   private:
   string StudentId;
   string StudentName;
   string StudentContactNumber;
   string BatchNo;
   string SemesterYear;//Fall,spring
   string RouteNumber;
   string StopName;
   static int TotalStudents;
   string CardNumber;

   Routes route;

   public:
   Student()
   {
      StudentId = " ";
      StudentName = " ";
      StudentContactNumber = " ";
      BatchNo = " ";
      SemesterYear = " ";//Fall,spring
```

```
    RouteNumber = " ";
    StopName = " ";
    CardNumber =" ";
  }
  Student(string StudentId,string StudentName,string
StudentContactNumber, string BatchNo,string SemesterYear,string
RouteNumber,string StopName,string CardNumber,string name,string
date,string Email,string number) : Person(name,date,Email,number)
  {
    this->StudentId = StudentId;
    this->StudentName = StudentName;
    this->StudentContactNumber = StudentContactNumber;
    this->BatchNo = BatchNo;
    this->SemesterYear = SemesterYear;
    this->RouteNumber = RouteNumber;
    this->StopName = StopName;
    this->CardNumber = CardNumber;
  }

  void SetAttributes(string StudentId,string StudentName,string
StudentContactNumber, string BatchNo,string SemesterYear,string
RouteNumber,string StopName,string CardNumber)
  {
```

```
    this->StudentId = StudentId;
    this->StudentName = StudentName;
    this->StudentContactNumber = StudentContactNumber;
    this->BatchNo = BatchNo;
    this->SemesterYear = SemesterYear;
    this->RouteNumber = RouteNumber;
    this->StopName = StopName;
    this->CardNumber = CardNumber;
}

void StudentRegisteration()
{
    string StudentId;
    cout << " Enter Your Fast Id like this(21k-0678) " << endl;
    cin >> StudentId;

    string StudentName;
    cout << " Enter Your Student Name " << endl;
    cin.ignore();
    getline(cin,StudentName);

    string StudentContactNumber;
```

```
cout << " Enter Student Contact Number " << endl;


cin >> StudentContactNumber;
string BatchNo;
cout << " Enter Batch No like 2022 " << endl;
cin >> BatchNo;


string SemesterYear;
cout << " Enter Semester(Fall/Spring)Year like Fall2024 ";
cin >> SemesterYear;


string RouteNumber;
cout << " Enter Route Number. Note Every Route Number start with (RouteNo) No represents The number of route like Route1 ,Route24 etc " << endl;
cin >> RouteNumber;




PaymentFee payment(21200);
int Routenumber = route.CheckRoute(RouteNumber);
if(Routenumber != -1)
{
```

```
        this->RouteNumber = RouteNumber;
        string RouteName = route.GetRouteName(RouteNumber);
        if(RouteName != " " && payment.GetStatus() != "NotPaid")
        {
            this->StopName = RouteName;
            cout << " Succesfully Registered " << endl;
            CardNumber = ("0"+ to_string(TotalStudents));

SetAttributes(StudentId,StudentName,StudentContactNumber,BatchN
o,SemesterYear,RouteNumber,RouteName,CardNumber);

            TotalStudents++;

        }
        else
        {
            cout << " Failed To Register " << endl;
        }


    }
    else
    {
```

```
        cout << " Route Not Found " << endl;
    }

}

string GetContactNumber() const
{
    return StudentContactNumber;
}

string GetName() const
{
    return StudentName;
}

static int GetTotalStudents()
{
    return TotalStudents;
}

string GetUserId() const
{
```

```
        return StudentId;
    }

    void DisplayInfo()
    {
        cout << " Student Details  Is  " << endl;
        cout << " StudentId  Is  " << StudentId <<  endl;
        cout << " Student Name is " << StudentName << endl;
        cout << " Student ContactNumber  Is  "<< StudentContactNumber
<< endl;
       cout << " Student BatchNo  Is  " << BatchNo << endl;
       cout << " Student SemesterYear  Is  " << SemesterYear << endl;
       cout <<"Route Number Is " <<  RouteNumber << "And Stop Name
Is "  << StopName << " And Card Number Is " << CardNumber << endl;
    }

    bool operator==(const Student& other) const
    {
       return this->StudentId == other.StudentId;
    }
};
```

```
int Student :: TotalStudents =0;

class Attendance
{
   private:
   int TotalAttendance;
   Student * student = nullptr;
   int * Attendancechecker;

   public:
   Attendance(int Totalstudents,Student * students)
   {

      Attendancechecker = new int [Totalstudents];
      TotalAttendance =0;
      student = students;
      for(int i =0; i < Totalstudents;i++)
      {
         Attendancechecker[i] =0;
      }
   }
```

```
void SetCard(string UserId,int Totalstudents)
{
   PaymentFee card;
   string status = card.GetStatus();
   for(int j =0; j < Totalstudents;j++)
   {
      if(student[j].GetUserId() == UserId)
      {
        TotalAttendance++;
        Attendancechecker[j] = TotalAttendance;
       }

   }



}

void DisplayInfo(int Totalstudents)
{
   for(int i =0; i < Totalstudents;i++)
```

```
      {
         cout << " Attendance is of " << i+1 << " Student is " << endl;

         cout << Attendancechecker[i] << endl;
         cout << " Name is " << student[i].GetName() << endl;

         cout << " Id is " << student[i].GetUserId()<<endl;

         cout << " Contact Number is " <<
student[i].GetContactNumber() << endl;

      }
   }


   ~Attendance()
   {

      delete [] Attendancechecker;
   }

};
```

```
class Teacher : public Person,public PaymentFee, public Routes
{
   private:
   double MonthlyFees;
   double Salary;

   string RouteNumber;



   public:
   Teacher(){}

   Teacher(string name,string date,string Email,string number,double
fees,string Routenumber,double salary) :
MonthlyFees(fees),Person(name,date,Email,number),RouteNumber(R
outenumber),PaymentFee(fees),Salary(salary)
   {}

   void DisplayInfo()
   {
      cout << endl;
      cout << " _Teacher Details Is_ " << endl;
```

```
    cout << " Teacher Name Is " << PersonName << endl;
    cout << " Teacher Email Is " << email << endl;
    cout << " Teacher Date Of 📅 Joining Is " << DateOfJoining <<
endl;
    cout << " Teacher Fees Is " << MonthlyFees << endl;
    cout << " Teacher Number Is " << phoneNumber << endl;
  }

  void TeacherRouteRegisteration()
  {
    PaymentFee payment(MonthlyFees);
    int result = PaymentFees(MonthlyFees);
    if(result !=0)
    {

      int Routenumber = CheckRoute(RouteNumber);
      if(Routenumber != -1)
      {
        string RouteName = GetRouteName(RouteNumber);
        if(RouteName != " " && payment.GetStatus() != "NotPaid")
        {
```

```
            cout << " Teacher Succesfully Registered " << endl;


          }
          else
          {
            cout << "Teacher Failed To Register " << endl;
          }



        }
        else
        {
         cout << " Route Not Found " << endl;
         }
      }


      else
      {
         cout << " Sorry Teacher But You Cannot Avail Transport Please
Clear Fees " << endl;
      }
   }
```

```
};


class StaffMembers : public Person,public PaymentFee,public Routes
{
   private:
   double salary;
   string RouteNumber;


   public:
   StaffMembers()
   {
      salary =0.0;
      RouteNumber = " ";
   }


   StaffMembers(double Salary,double fees,string name,string
date,string Email,string number,string Route) :
salary(Salary),Person(name,date,Email,number),RouteNumber(Route)
,PaymentFee(fees)
   {
```

```
}

void DisplayInfo()
{
   cout << endl;
   cout << " _Staff Info Is_ " << endl;
   cout << " Staff Name Is " << PersonName << endl;
   cout << " Staff Email Is " << email << endl;
   cout << "  Staff Date Of 📅 Joining Is " << DateOfJoining << endl;
   cout << " Stafff Fees Is " << salary << endl;
   cout << " Staff Number Is " << phoneNumber << endl;
}


void StaffRouteRegisteration()
{
   PaymentFee payment(salary);
   int result = payment.PaymentFees(salary);

   if(result !=0)
   {
     cout << " Status " << payment.GetStatus() << endl;
```

```
  int Routenumber = CheckRoute(RouteNumber);
  if(Routenumber != -1)
  {
    string RouteName = GetRouteName(RouteNumber);
    if(RouteName != " " && payment.GetStatus() != "NotPaid")
    {

      cout << "Staff Succesfully Registered " << endl;


    }
    else
    {
      cout << "Staff Failed To Register " << endl;
    }



  }
  else
  {
   cout << " Route Not Found " << endl;
   }
}
```

```
        else
        {
            cout << " Sorry Staff you  Cannot Avail Transport Please Clear Fees " << endl;
        }
    }

};

int main()
{


   cout << " Welcome To Fast Bus Transportation System " << endl;
   cout << endl;

   int TotalStudents;
   cout << " Enter Total Students who want to make there Card " << endl;
   cin >> TotalStudents;

   Student * student = new Student[TotalStudents];
```

```
  for(int i =0; i < TotalStudents;i++)
  {
     student[i].StudentRegisteration();

  }

  Person *  person1 = new Person("Ali", "01-01-2023", "ali@fast.edu",
"03001234567");
  Person * person2 = new Person("li", "01-01-2023", "ali@fast.edu",
"03001234567");

  string userid;
  cout << " Enter User id whose Attendance is to be displayed " <<
endl;
  cin >> userid;

  Attendance attendance(TotalStudents,student);
  attendance.SetCard(userid,TotalStudents);
  attendance.DisplayInfo(TotalStudents);

  Teacher  teacher1("John Doe", "01-02-2023", "johndoe@fast.edu",
"03001234567", 5000, "Route3", 100000);
```

```
Person * person; // For Dynamic/LateBinding
person = &teacher1;

cout << endl;
teacher1.TeacherRouteRegisteration();
person->DisplayInfo();

cout << endl;

StaffMembers
staffmember1(34096,120,"Qasim","3-45-2099","12@nu","34","Route
24");
cout << endl;

Person * Pes; // For Dynamics / Late Binding
Pes = & staffmember1;

staffmember1.StaffRouteRegisteration();
Pes->DisplayInfo();

cout << endl;
cout <<" Operator Checking Outcomes Are "<< endl;
```

```
 Routes route1, route2;
 if (route1 == route2)
 {
  cout << "Both routes are the same." << endl;
 }
 else
 {
   cout << "Routes are different." << endl;
}

cout << endl;



if (person1 == person2)
{
 cout << "Both persons are the same." << endl;
 person2->DisplayInfo();
}
else
{
 cout << "Persons are different." << endl;
```

```
    person1->DisplayInfo();
    person2->DisplayInfo();
   }



   if (student[0] == student[1])
   {
    cout << "Both Students are the same." << endl;
   }
   else
   {
    cout << "Students are different." << endl;

   }



   delete [] student;

    return 0;
}
```

OUTPUTQ1:

main.cpp

```cpp
//24K-0559 BAZIL-UDDIN-KHAN
#include <iostream>
#include<cstring>
#include<string>
using namespace std;

class Person
{
    protected:
    string PersonName;
    string DateOfJoining;
    string email;
    string phoneNumber;

    public:
    Person(){}

    Person(string name,string date,string Email,string number) : PersonName(name
        ),DateOfJoining(date),email(Email),phoneNumber(number)
    {}

    virtual  void DisplayInfo()
    {
        cout << " Person Details Are " << endl;
        cout << "Person Name is = " << PersonName << endl;
        cout << "Person Date Of Joining  is = " << DateOfJoining << endl;
        cout << "Person Email is = " << email << endl;
        cout << "Person Phone Number  is = " << phoneNumber << endl;

    }

    bool operator==(const Person& other) const
    {
    return this->email == other.email;
    }
```

Output

```
Welcome To Fast Bus Transportation System

 Enter Total Students who want to make there Card
3
 Enter Your Fast Id like this(21k-0678)
21k-786
 Enter Your Student Name
Qasim
 Enter Student Contact Number
123456789
 Enter Batch No like 2022
2021
 Enter Semester(Fall/Spring)Year like Fall2024 Fall2021
 Enter Route Number. Note Every Route Number start with (RouteNo) No represents The number of
     route like Route1 ,Route24 etc
Route25
 Succesfully Registered
 Enter Your Fast Id like this(21k-0678)
23K-5437
 Enter Your Student Name
Ali
 Enter Student Contact Number
123444444
 Enter Batch No like 2022
2023
 Enter Semester(Fall/Spring)Year like Fall2024 Fall2023
 Enter Route Number. Note Every Route Number start with (RouteNo) No represents The number of
     route like Route1 ,Route24 etc
Route4B
 Succesfully Registered
 Enter Your Fast Id like this(21k-0678)
24k-6789
 Enter Your Student Name
Wasim
 Enter Student Contact Number
1345678
```

main.cpp

```cpp
//24K-0559 BAZIL-UDDIN-KHAN
#include <iostream>
#include<cstring>
#include<string>
using namespace std;

class Person
{
    protected:
    string PersonName;
    string DateOfJoining;
    string email;
    string phoneNumber;

    public:
    Person(){}

    Person(string name,string date,string Email,string number) : PersonName(name
        ),DateOfJoining(date),email(Email),phoneNumber(number)
    {}

    virtual  void DisplayInfo()
    {
        cout << " Person Details Are " << endl;
        cout << "Person Name is = " << PersonName << endl;
        cout << "Person Date Of Joining  is = " << DateOfJoining << endl;
        cout << "Person Email is = " << email << endl;
        cout << "Person Phone Number  is = " << phoneNumber << endl;

    }

    bool operator==(const Person& other) const
    {
    return this->email == other.email;
    }
```

Output

```
 Enter Student Contact Number
1345678
 Enter Batch No like 2022
2024
 Enter Semester(Fall/Spring)Year like Fall2024 Fall2024
 Enter Route Number. Note Every Route Number start with (RouteNo) No represents The number of
     route like Route1 ,Route24 etc
Route7
 Succesfully Registered
 Enter User id whose Attendance is to be displayed
23K-5437
 Attendance is of 1 Student is
0
 Name is Qasim
 Id is 21k-786
 Contact Number is 123456789
 Attendance is of 2 Student is
1
 Name is Ali
 Id is 23K-5437
 Contact Number is 123444444
 Attendance is of 3 Student is
0
 Name is Wasim
 Id is 24k-6789
 Contact Number is 1345678

 Payment Sussfully made
 Teacher Succesfully Registered

_Teacher Details Is_
 Teacher Name Is John Doe
 Teacher Email Is johndoe@fast.edu
 Teacher Date Of 🗓 Joining Is 01-02-2023
 Teacher Fees Is 5000
```

main.cpp

Share Run

```cpp
//24K-0559 BAZIL-UDDIN-KHAN
#include <iostream>
#include<cstring>
#include<string>
using namespace std;

class Person
{
    protected:
    string PersonName;
    string DateOfJoining;
    string email;
    string phoneNumber;

    public:
    Person(){}

    Person(string name,string date,string Email,string number) : PersonName(name
        ),DateOfJoining(date),email(Email),phoneNumber(number)
    {}

    virtual  void DisplayInfo()
    {
        cout << " Person Details Are " << endl;
        cout << "Person Name is = " << PersonName << endl;
        cout << "Person Date Of Joining  is = " << DateOfJoining << endl;
        cout << "Person Email is = " << email << endl;
        cout << "Person Phone Number  is = " << phoneNumber << endl;

    }

    bool operator==(const Person& other) const
    {
    return this->email == other.email;
    }
```

Output

```
_Teacher Details Is_
Teacher Name Is John Doe
Teacher Email Is johndoe@fast.edu
Teacher Date Of  Joining Is 01-02-2023
Teacher Fees Is 5000
Teacher Number Is 03001234567


Payment Sussfully made
Status Paid
Staff Succesfully Registered

_Staff Info Is_
Staff Name Is Qasim
Staff Email Is 12@nu
 Staff Date Of  Joining Is 3-45-2099
Stafff Fees Is 34096
Staff Number Is 34

Operator Checking Outcomes Are
Both routes are the same.

Persons are different.
 Person Details Are
Person Name is = Ali
Person Date Of Joining  is = 01-01-2023
Person Email is = ali@fast.edu
Person Phone Number  is = 03001234567
 Person Details Are
Person Name is = li
Person Date Of Joining  is = 01-01-2023
Person Email is = ali@fast.edu
Person Phone Number  is = 03001234567
Students are different.
```

QUESTION2).

CODE:

```
//24k-0559 Bazil-Uddin-Khan
#include <iostream>
using namespace std;


class Ghost
{
   protected:
   string PlayerName;
   int Scarelevel;
```

```
public:
Ghost()
{
   PlayerName = " ";
   Scarelevel =0;
}

Ghost(string playername, int scarelevel) : PlayerName(playername), Scarelevel(scarelevel)
{}

int GetScareLevel()
{
   return Scarelevel;
}

virtual void PerformHaunting()
{
   cout << " Player Name " << PlayerName << " Should Haunt Visitors " << endl;

}
```

```
   string GetName()
   {
      return PlayerName;
   }



  friend ostream& operator << (ostream& out,Ghost &ghost);


  Ghost operator+(Ghost & ghost)
  {
     return Ghost(PlayerName + "&" + ghost.PlayerName , Scarelevel +
ghost.Scarelevel);


  }


};


ostream& operator << (ostream& out,Ghost &ghost)
{
      out << " Ghost Player Name is  "  << ghost.PlayerName << " Ghost
Scare Level Is  " << ghost.Scarelevel << endl;
      return out;
```

```
}


class Poltergeists : virtual public Ghost
{
   public:
   Poltergeists()
   {}

   Poltergeists(string playername, int scarelevel ) :
Ghost(playername,scarelevel)
   {}



    void PerformHaunting()
   {
      cout << "Player = " << PlayerName << " is  Poltergeists who Moves
Objects " << endl;
   }
};

class Banshees : virtual public Ghost
```

```
{
   public:
   Banshees()
   {}

   Banshees(string playername, int scarelevel ) :
Ghost(playername,scarelevel)
   {}

   void PerformHaunting()
   {
      cout << " Player = " << PlayerName << " who is (Banshees)
Scream's Loudly " << endl;
   }
};



class ShadowGhosts : virtual public Ghost
{
   public:
   ShadowGhosts()
   {}
```

```
  ShadowGhosts(string playername, int scarelevel ) :
Ghost(playername,scarelevel)
  {}

  void PerformHaunting()
  {
     cout << "Player Name " << PlayerName << " is (ShadowGhosts)
who is to Whispher Creeply  " << endl;
  }
};

class HybridGhost : public ShadowGhosts, public Poltergeists
{
  public:
  HybridGhost()
  {}

  HybridGhost(string playername, int scarelevel ) :
Ghost(playername,scarelevel),ShadowGhosts(playername,scarelevel),
Poltergeists(playername,scarelevel)
  {}
```

```
    void PerformHaunting()
    {
        ShadowGhosts :: PerformHaunting();
        Poltergeists :: PerformHaunting();
    }

};


class HauntedHouse
{
    string HauntedHouseName;
    Ghost ** ghost;
    int count;

    public:
    HauntedHouse()
    {
        HauntedHouseName = " ";
        count =0;
```

```
    }
    HauntedHouse(string housename) :
HauntedHouseName(housename),count(0)
    {
      ghost = new Ghost*[100];
    }

    void AddGhost(Ghost* ghosts)
    {
       if (count < 100)
       {
          ghost[count] = ghosts;
          count++;
       }
       else
       {
          cout << " Sorry!. Haunted house is full!" << endl;
       }
    }

    void StartHaunting()
```

```
    {
        cout << "Haunted House Name is =  " << HauntedHouseName <<
endl;
        for (int i = 0; i < count; i++)
        {
            ghost[i]->PerformHaunting();
        }

    }

    string GetHouseName()
    {
        return HauntedHouseName;
    }




};

class Visitor
{
    private:
```

```
   string VisitorName;
   int BraveryLevel;

   public:
   Visitor()
   {
      VisitorName = " ";
      BraveryLevel = 0;
   }

   Visitor(string name, int level) :
VisitorName(name),BraveryLevel(level)
   {}

   friend void Visit(Visitor visitors[],HauntedHouse & haunted,Ghost  *
ghost1);

   string GetName()
   {
      return VisitorName;
   }
```

```
    int GetLevel()
    {
        return BraveryLevel;
    }



};

void Visit(Visitor visitors[], HauntedHouse &haunted, Ghost *ghosts[])
{
    cout << " Visitors Entering Our Haunted House : " <<
haunted.GetHouseName() << endl;

    for (int i = 0; i < 3; i++)
    {
        cout << visitors[i].GetName() << "  is entering the haunted house "
<< endl;



        string braveryRange;
        if (visitors[i].GetLevel() >= 1 && visitors[i].GetLevel() <= 4)
        {
            braveryRange = "Cowardly";
```

```
    }

    else if (visitors[i].GetLevel() >= 5 && visitors[i].GetLevel() <= 7)
    {
       braveryRange = "Average";
    }

    else if (visitors[i].GetLevel() >= 8 && visitors[i].GetLevel() <= 10)
    {
       braveryRange = "Fearless";
    }

    cout << visitors[i].GetName() << " is " << braveryRange << endl;

    for (int j = 0; j < 4; j++)
    {
       int scareLevel = ghosts[j]->GetScareLevel();
       cout << ghosts[j]->GetName() << " is trying to scare them with a
scare level of " << scareLevel  << endl;

       if (scareLevel < 5)
       {
```

```
            cout << visitors[i].GetName() << " Ha Ha he says .is that all
you got ?." << endl;
        }
        else if (scareLevel >= 5 && scareLevel <= 7)
        {
            cout << visitors[i].GetName() << " Speaking with  shaky voice
I am shivering. " << endl;
        }
        else
        {
            cout << "Ah help " << visitors[i].GetName() << " is running
away in fear " << endl;
        }
    }

    cout << endl;
  }
}

int main()
{
```

```
HauntedHouse house1("Borley Rectory");

Ghost * ghost [4]
=
{
new Poltergeists("Qasim", rand() % 10 + 1),
new Banshees("Wasim", rand() % 10 + 1),
new ShadowGhosts("Ali", rand() % 10 + 1),
new HybridGhost("Farteen", rand() % 10 + 1) };

cout << " Ghost Info " << endl;
for (int i = 0; i < 4; i++)
{
cout << *ghost[i];
}

Ghost *upgradedGhost = new Ghost(*ghost[0] + *ghost[1]);

cout << " New Upgraded Ghost Is  ";
cout << *upgradedGhost;
```

```
house1.AddGhost(ghost[0]);
house1.AddGhost(ghost[1]);
house1.AddGhost(ghost[2]);
house1.AddGhost(ghost[3]);

cout << endl;
cout << " _House 1  Visitors_ " << endl;
cout << endl;


Visitor visitors1[3] = {{"Saffora",6},{"Quba",2},{"Was",9}};

Visit(visitors1, house1, ghost);



HauntedHouse house2("Amit valley");

cout << endl;
cout << " _House 2 Visitors_ " << endl;

house2.AddGhost(ghost[0]);
```

```
house2.AddGhost(ghost[1]);
house2.AddGhost(ghost[2]);
house2.AddGhost(ghost[3]);

Visitor visitors2[3] = {{"Saff",1},{"Qua",9},{"Ws",5}};

Visit(visitors2, house2, ghost);


HauntedHouse house3("Mystery");
cout << endl;
cout << " _House 3 Visitors_ " << endl;
house3.AddGhost(ghost[0]);
house3.AddGhost(ghost[1]);
house3.AddGhost(ghost[2]);
house3.AddGhost(ghost[3]);

Visitor visitors3[3] = {{"Myste",6},{"Qua",2},{"Ws",9}};

Visit(visitors3, house3, ghost);

return 0;
```

}

OUTPUTQ2:

main.cpp

```cpp
//24k-0559 Bazil-Uddin-Khan
#include <iostream>
using namespace std;

class Ghost
{
    protected:
    string PlayerName;
    int Scarelevel;

    public:
    Ghost()
    {
        PlayerName = " ";
        Scarelevel =0;
    }

    Ghost(string playername, int scarelevel) : PlayerName(playername), Scarelevel
        (scarelevel)
    {}

    int GetScareLevel()
    {
        return Scarelevel;
    }

    virtual void PerformHaunting()
    {
        cout << " Player Name " << PlayerName << " Should Haunt Visitors " << endl;

    }

    string GetName()
    {
        return PlayerName;
```

Output

```
Ghost Info
Ghost Player Name is  Qasim Ghost Scare Level Is  4
Ghost Player Name is  Wasim Ghost Scare Level Is  7
Ghost Player Name is  Ali Ghost Scare Level Is  8
Ghost Player Name is  Farteen Ghost Scare Level Is  6
New Upgraded Ghost Is   Ghost Player Name is  Qasim&Wasim Ghost Scare Level Is  11

_House 1  Visitors_

 Visitors Entering Our Haunted House : Borley Rectory
Saffora  is entering the haunted house
Saffora is Average
Qasim is trying to scare them with a scare level of 4
Saffora Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Saffora Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Saffora is running away in fear
Farteen is trying to scare them with a scare level of 6
Saffora Speaking with  shaky voice I am shivering.

Quba  is entering the haunted house
Quba is Cowardly
Qasim is trying to scare them with a scare level of 4
Quba Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Quba Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Quba is running away in fear
Farteen is trying to scare them with a scare level of 6
Quba Speaking with  shaky voice I am shivering.

Was  is entering the haunted house
Was is Fearless
Qasim is trying to scare them with a scare level of 4
Was Ha Ha he says  is that all you got ?
```

main.cpp

```cpp
//24k-0559 Bazil-Uddin-Khan
#include <iostream>
using namespace std;

class Ghost
{
    protected:
    string PlayerName;
    int Scarelevel;

    public:
    Ghost()
    {
        PlayerName = " ";
        Scarelevel =0;
    }

    Ghost(string playername, int scarelevel) : PlayerName(playername), Scarelevel
        (scarelevel)
    {}

    int GetScareLevel()
    {
        return Scarelevel;
    }

    virtual void PerformHaunting()
    {
        cout << " Player Name " << PlayerName << " Should Haunt Visitors " << endl;

    }

    string GetName()
    {
        return PlayerName;
```

Output

```
Was  is entering the haunted house
Was is Fearless
Qasim is trying to scare them with a scare level of 4
Was Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Was Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Was is running away in fear
Farteen is trying to scare them with a scare level of 6
Was Speaking with  shaky voice I am shivering.


 _House 2 Visitors_
 Visitors Entering Our Haunted House : Amit valley
Saff  is entering the haunted house
Saff is Cowardly
Qasim is trying to scare them with a scare level of 4
Saff Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Saff Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Saff is running away in fear
Farteen is trying to scare them with a scare level of 6
Saff Speaking with  shaky voice I am shivering.

Qua  is entering the haunted house
Qua is Fearless
Qasim is trying to scare them with a scare level of 4
Qua Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Qua Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Qua is running away in fear
Farteen is trying to scare them with a scare level of 6
Qua Speaking with  shaky voice I am shivering.
```

main.cpp

Share Run

```cpp
//24k-0559 Bazil-Uddin-Khan
#include <iostream>
using namespace std;

class Ghost
{
    protected:
    string PlayerName;
    int Scarelevel;

    public:
    Ghost()
    {
        PlayerName = " ";
        Scarelevel =0;
    }

    Ghost(string playername, int scarelevel) : PlayerName(playername), Scarelevel
        (scarelevel)
    {}

    int GetScareLevel()
    {
        return Scarelevel;
    }

    virtual void PerformHaunting()
    {
        cout << " Player Name " << PlayerName << " Should Haunt Visitors " << endl;

    }

    string GetName()
    {
        return PlayerName;
    }
```

Output

```
Ws  is entering the haunted house
Ws is Average
Qasim is trying to scare them with a scare level of 4
Ws Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Ws Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Ws is running away in fear
Farteen is trying to scare them with a scare level of 6
Ws Speaking with  shaky voice I am shivering.


 _House 3 Visitors_
 Visitors Entering Our Haunted House : Mystery
Myste  is entering the haunted house
Myste is Average
Qasim is trying to scare them with a scare level of 4
Myste Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Myste Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Myste is running away in fear
Farteen is trying to scare them with a scare level of 6
Myste Speaking with  shaky voice I am shivering.

Qua  is entering the haunted house
Qua is Cowardly
Qasim is trying to scare them with a scare level of 4
Qua Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Qua Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Qua is running away in fear
Farteen is trying to scare them with a scare level of 6
Qua Speaking with  shaky voice I am shivering.
```

main.cpp

Share Run

```cpp
//24k-0559 Bazil-Uddin-Khan
#include <iostream>
using namespace std;

class Ghost
{
    protected:
    string PlayerName;
    int Scarelevel;

    public:
    Ghost()
    {
        PlayerName = " ";
        Scarelevel =0;
    }

    Ghost(string playername, int scarelevel) : PlayerName(playername), Scarelevel
        (scarelevel)
    {}

    int GetScareLevel()
    {
        return Scarelevel;
    }

    virtual void PerformHaunting()
    {
        cout << " Player Name " << PlayerName << " Should Haunt Visitors " << endl;

    }

    string GetName()
    {
        return PlayerName;
    }
```

Output

```
Myste  is entering the haunted house
Myste is Average
Qasim is trying to scare them with a scare level of 4
Myste Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Myste Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Myste is running away in fear
Farteen is trying to scare them with a scare level of 6
Myste Speaking with  shaky voice I am shivering.

Qua  is entering the haunted house
Qua is Cowardly
Qasim is trying to scare them with a scare level of 4
Qua Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Qua Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Qua is running away in fear
Farteen is trying to scare them with a scare level of 6
Qua Speaking with  shaky voice I am shivering.

Ws  is entering the haunted house
Ws is Fearless
Qasim is trying to scare them with a scare level of 4
Ws Ha Ha he says .is that all you got ?.
Wasim is trying to scare them with a scare level of 7
Ws Speaking with  shaky voice I am shivering.
Ali is trying to scare them with a scare level of 8
Ah help Ws is running away in fear
Farteen is trying to scare them with a scare level of 6
Ws Speaking with  shaky voice I am shivering.


=== Code Execution Successful ===
```

QUESTION3).

CODE:

```
// 24k-0559 BAZIL-UDDIN-KHAN
#include <iostream>
using namespace std;

class Vehicle
{
   private:
   double Capacity;
   double EnergyConsumption;
   double AverageSpeed;



   static  int Activedelieveries;


   protected:
   string vehicleID;


   string Routes[6] =
{"Clifton","Millinium","Saddar","Nazimabad","Gulshan","Malir"};
```

```
   double EstimatedTimeInHour[6] = {3.4,4,2,2,4,1.5};
   double Distance[6] ={2,5,3,7,8,9};
   string FoodAvailable[10] = {"Biryani", "Nihari", "Haleem", "Paya",
"Chapli Kebab", "Sajji", "Karahi", "Aloo Paratha", "Chana Chaat",
"Zarda"};
   double Prices[10] = {
      300.0, 350.0, 190.0, 350.0, 0.0, 1430.0, 960.0, 350.0, 490.0, 800};


   public:
   Vehicle(){vehicleID = " ";}


   Vehicle(string Id,double capacity,
   double Consumption,
   double Speed) :
vehicleID(Id),AverageSpeed(Speed),EnergyConsumption(Consumption
),Capacity(capacity)
   {}
   string GetvehicleID()
   {
      return vehicleID;


   }
   double GetAverageSpeed()
```

```
{
   return AverageSpeed;
}
double EstimatedTimeDelievery()
{
   cout << " _Locations Available For Delivery Are_ " << endl;
   for(int y =0; y<6;y++)
   {
      cout << " " << Routes[y] << "  " << endl;
   }

   string Choice;
   cout << " Enter Your Location For Order " << endl;
   cin >> Choice;

   int Index = -1;
   for(int u =0; u<6; u++)
   {
      if(Routes[u] == Choice)
      {
         Index = u;
      }
```

```
        }
        if(Index != -1)
        {
           double TimeReq = (Distance[Index]
*EstimatedTimeInHour[Index] - Distance[Index]);
           cout << " Time Of Delievery Is  = " << TimeReq << endl;
           return TimeReq;

        }

        else
        {
           cout << " Sorry We Donot Provide Seruve Here " << endl;
           return 0;
        }


    }

  friend bool operator==(const Vehicle& v1, const Vehicle& v2);
```

```
void OptimumDeilieveryRoute(double TimeReq )
{

   double Optimum = TimeReq;
   int index =0;
   for(int y =1; y < 6; y++)
   {
      if(Optimum<=EstimatedTimeInHour[y])
      {
         Optimum = EstimatedTimeInHour[y];
         index = y;
      }
   }


   cout << " The Optimum Delievery Route For Deilevery will Be Through  " << Routes[index] << endl;
   Activedelieveries++;
   cout << "Active Delievers Are " << Activedelieveries << endl;
```

```
}

virtual void VehicleMovement()
{
    string Order;
    cout << " __Orders Available Are__ : " << endl;
    for(int i =0; i <10; i++)
    {
       cout << " " <<  FoodAvailable[i] << " " << endl;
    }
    double TotalBill = 0.0;
while(1)
{
     cout << " Do You Want To Place Order ? (Yes/No) " << endl;
    cin >> Order;
    if(Order == "Yes")
    {

          string Choice;
          cout << " Enter Meal Name " << endl;
          cin >> Choice;
```

```
        for(int i =0; i < 10;i++)
        {
           if(Choice ==FoodAvailable[i])
           {

              TotalBill = TotalBill + Prices[i];
           }
        }
    }




    else
    {
       cout << " Thanks For Visiting Our System " << endl;
       cout << " Total Bill Is " << TotalBill << endl;
       break;
    }
}
}
```

```
};


int Vehicle :: Activedelieveries =0;


bool operator==(const Vehicle& v1, const Vehicle& v2) {
   return (v1.AverageSpeed == v2.AverageSpeed) && (v1.Capacity == v2.Capacity) && (v1.EnergyConsumption == v2.EnergyConsumption);
}


class RamazanDrone : public Vehicle
{
   private:


   protected:


   double Moisture[6] = {5,6,2,3,4,7};
   double DroneSpeed[3] = {45,75,100};
```

```
public:
RamazanDrone()
{}

RamazanDrone(string Id,double capacity,
double Consumption,
double Speed) : Vehicle(Id,capacity,
 Consumption, Speed)
{}

void VehicleMovement()
{
  Vehicle :: VehicleMovement();
  double TimeReq = EstimatedTimeDelievery();
  OptimumDeilieveryRoute(TimeReq);

  double SmallestMoisture = Moisture[0];
  int Index = -1;
  for(int y =1;y<6;y++)
   {
```

```
        if(Moisture[y] < SmallestMoisture)
        {
            SmallestMoisture = Moisture[y];
            Index = y;
        }
    }

    if(Index != -1)
    {
        cout << " Drone Taking The Route Where Moisture Lowest Is = "
<< Moisture[Index] << " Location is = " << Routes[Index] << endl;

    }

  }


};


class RamazanTimeShip : public Vehicle
{
```

```
protected:

private:
double Capacity;
double EnergyConsumption;
double AverageSpeed;
double Protocol[2] = {2,8};
string Protocolindex[2] = {
   "Time Taken","Ratings Provied Out Bw(1-10) "};


public:
RamazanTimeShip()
{}

RamazanTimeShip(string Id,double capacity,
double Consumption,
double Speed) : Vehicle(Id,capacity,Consumption,Speed)
{}

void VehicleMovement()
{
```

```
    cout << "Vehicle Id Of " << vehicleID <<  "is Goining On A Visit In History " << endl;
    Vehicle :: VehicleMovement();

    double timereq = EstimatedTimeDelievery();
    if(timereq !=0)
    {
       OptimumDeilieveryRoute(timereq);
       double rating;
       cout << " Enter Rating (1-10) " << endl;
       cin >> rating;

       if(timereq <=2 && rating >=8)
       {
          Protocol[0] = timereq;
          Protocol[1] = rating;
          cout << " Time Machine Succesfully Toured History And Met Protocols With Updated Recent History " << endl;
       }
       else
       {
          cout << " Meet History Standards Now!. " << endl;
```

```
        }

      }

   }

};

class RamazanHyperPod : public Vehicle
{
   private:
   double Capacity;
   double EnergyConsumption;
   double AverageSpeed;

   protected:
   double TunnelCapacityInMeter[6] =
   {12,32,47,34,23,10};

   public:
   RamazanHyperPod()
   {}
```

```
RamazanHyperPod(string Id,double capacity,
double Consumption,
double Speed) : Vehicle(Id,capacity,Consumption,Speed)
{}

void VehicleMovement()
{

   double TimeReq = EstimatedTimeDelievery();
   if(TimeReq != 0)
   {
     OptimumDeilieveryRoute(TimeReq);

     int index =0;
    double HighestCapacity = TunnelCapacityInMeter[0];
     for(int y =1;y<6;y++)
    {
       if(HighestCapacity < TunnelCapacityInMeter[y] )
       {
          HighestCapacity =  TunnelCapacityInMeter[y];
          index = y;
```

```
            }
        }

        cout << " From The Location " << Routes[index] << " And The
Tunnel Has This Capaciry " << TunnelCapacityInMeter[index] << endl;
      }
    }

};

class OperatorControlPanel
{
public:
   void command(string action, int packageID)
   {
      cout << "Executing " << action << " for package " << packageID <<
" normally." << endl;
   }

   void command(string action, int packageID, string urgencyLevel) {
      cout << "Executing " << action << " for package " << packageID <<
" with urgency: " << urgencyLevel << endl;
      }
```

```
};


class AiConflictResolutionSystem
{
public:
   static Vehicle resolveConflict(Vehicle v1, Vehicle v2)
   {
      if (v1 == v2)
      {
         cout << "Both vehicles are efficient.But Assigning first available vehicle." << endl;
         return v1;

      }
      else if (v1.GetAverageSpeed() > v2.GetAverageSpeed())
      {
         cout << " Vehicle 1 is faster. Selecting v1" << endl;
         return v1;

      }
      else
```

```
        {
            cout << "Vehicle 2 is more efficient. Selecting v2" << endl;
            return v2;
        }
    }
};



int main()
{

    RamazanDrone drone("DroneX", 10, 85, 120);
    RamazanTimeShip timeShip("TimeShipY", 30, 90, 80);
    RamazanHyperPod hyperPod("HyperPodZ", 100, 75, 150);

   cout << " Ramazan Drome " << endl;
    double droneTime = drone.EstimatedTimeDelievery();
    drone.OptimumDeilieveryRoute(droneTime);

    cout << " Time Ship " << endl;
    double timeShipTime = timeShip.EstimatedTimeDelievery();
```

```
    timeShip.OptimumDeilieveryRoute(timeShipTime);

    cout << " HyperPod " << endl;
    double hyperPodTime = hyperPod.EstimatedTimeDelievery();
    hyperPod.OptimumDeilieveryRoute(hyperPodTime);


   cout << endl;
   cout << endl;
   cout << " Command Deciding " << endl;
    OperatorControlPanel panel;
    panel.command("Deliver", 101);
    panel.command("Deliver", 102, "Urgent");


    cout << "  AI Conflict Resolution Test  " << endl;
    Vehicle vehicle1("DroneX", 10, 85, 120);
    Vehicle vehicle2("HyperPodY", 50, 95, 120);


    Vehicle selectedVehicle =
AiConflictResolutionSystem::resolveConflict(vehicle1, vehicle2);
```

```
    cout << " Ai is selecteing the following vehicle for delivery:" << endl;
    cout << "Vehicle ID: " << selectedVehicle.GetvehicleID() << ", Speed:
" << selectedVehicle.GetAverageSpeed() << endl;



    cout << "  Vehicle Movements "  << endl;
    drone.VehicleMovement();
    timeShip.VehicleMovement();
    hyperPod.VehicleMovement();

    return 0;
}
```

OUTPUT Q3:

main.cpp

Share Run

```cpp
// 24k-0559 BAZIL-UDDIN-KHAN
#include <iostream>
using namespace std;

class Vehicle
{
    private:
    double Capacity;
    double EnergyConsumption;
    double AverageSpeed;



    static  int Activedelieveries;

    protected:
    string vehicleID;

    string Routes[6] = {"Clifton","Millinium","Saddar","Nazimabad","Gulshan","Malir"};
    double EstimatedTimeInHour[6] = {3.4,4,2,2,4,1.5};
    double Distance[6] ={2.5,3,7.8,9};
    string FoodAvailable[10] = {"Biryani", "Nihari", "Haleem", "Paya", "Chapli Kebab",
        "Sajji", "Karahi", "Aloo Paratha", "Chana Chaat", "Zarda"};
    double Prices[10] = {
        300.0, 350.0, 190.0, 350.0, 0.0, 1430.0, 960.0, 350.0, 490.0, 800};

    public:
    Vehicle(){vehicleID = " ";}

    Vehicle(string Id,double capacity,
    double Consumption,
    double Speed) : vehicleID(Id),AverageSpeed(Speed),EnergyConsumption(Consumption
        ),Capacity(capacity)
    {}
    string GetvehicleID()
    {
```

Output

```
 Ramazan Drome
 _Locations Available For Delivery Are_
 Clifton
 Millinium
 Saddar
 Nazimabad
 Gulshan
 Malir
 Enter Your Location For Order
Clifton
 Time Of Delievery Is  = 4.8
 The Optimum Delievery Route For Deilevery will Be Through  Clifton
Active Delievers Are 1
 Time Ship
 _Locations Available For Delivery Are_
 Clifton
 Millinium
 Saddar
 Nazimabad
 Gulshan
 Malir
 Enter Your Location For Order
Malir
 Time Of Delievery Is  = 4.5
 The Optimum Delievery Route For Deilevery will Be Through  Clifton
Active Delievers Are 2
 HyperPod
 _Locations Available For Delivery Are_
 Clifton
 Millinium
 Saddar
 Nazimabad
 Gulshan
 Malir
 Enter Your Location For Order
Gulshan
```

main.cpp

Share Run

```cpp
// 24k-0559 BAZIL-UDDIN-KHAN
#include <iostream>
using namespace std;

class Vehicle
{
    private:
    double Capacity;
    double EnergyConsumption;
    double AverageSpeed;



    static  int Activedelieveries;

    protected:
    string vehicleID;

    string Routes[6] = {"Clifton","Millinium","Saddar","Nazimabad","Gulshan","Malir"};
    double EstimatedTimeInHour[6] = {3.4,4,2,2,4,1.5};
    double Distance[6] ={2.5,3,7.8,9};
    string FoodAvailable[10] = {"Biryani", "Nihari", "Haleem", "Paya", "Chapli Kebab",
        "Sajji", "Karahi", "Aloo Paratha", "Chana Chaat", "Zarda"};
    double Prices[10] = {
        300.0, 350.0, 190.0, 350.0, 0.0, 1430.0, 960.0, 350.0, 490.0, 800};

    public:
    Vehicle(){vehicleID = " ";}

    Vehicle(string Id,double capacity,
    double Consumption,
    double Speed) : vehicleID(Id),AverageSpeed(Speed),EnergyConsumption(Consumption
        ),Capacity(capacity)
    {}
    string GetvehicleID()
    {
```

Output

```
 Enter Your Location For Order
Gulshan
 Time Of Delievery Is  = 24
 The Optimum Delievery Route For Deilevery will Be Through  Clifton
Active Delievers Are 3


 Command Deciding
Executing Deliver for package 101 normally.
Executing Deliver for package 102 with urgency: Urgent
  AI Conflict Resolution Test
Vehicle 2 is more efficient. Selecting v2
 Ai is selecteing the following vehicle for delivery:
Vehicle ID: HyperPodY, Speed: 120
  Vehicle Movements
 __Orders Available Are__ :
 Biryani
 Nihari
 Haleem
 Paya
 Chapli Kebab
 Sajji
 Karahi
 Aloo Paratha
 Chana Chaat
 Zarda
 Do You Want To Place Order ? (Yes/No)
Yes
 Enter Meal Name
Zarda
 Do You Want To Place Order ? (Yes/No)
Yes
 Enter Meal Name
Biryani
 Do You Want To Place Order ? (Yes/No)
No
```

main.cpp

Share Run

```cpp
// 24k-0559 BAZIL-UDDIN-KHAN
#include <iostream>
using namespace std;

class Vehicle
{
    private:
    double Capacity;
    double EnergyConsumption;
    double AverageSpeed;



    static  int Activedelieveries;

    protected:
    string vehicleID;

    string Routes[6] = {"Clifton","Millinium","Saddar","Nazimabad","Gulshan","Malir"};
    double EstimatedTimeInHour[6] = {3.4,4,2,2,4,1.5};
    double Distance[6] ={2,5,3,7,8,9};
    string FoodAvailable[10] = {"Biryani", "Nihari", "Haleem", "Paya", "Chapli Kebab",
        "Sajji", "Karahi", "Aloo Paratha", "Chana Chaat", "Zarda"};
    double Prices[10] = {
        300.0, 350.0, 190.0, 350.0, 0.0, 1430.0, 960.0, 350.0, 490.0, 800};

    public:
    Vehicle(){vehicleID = " ";}

    Vehicle(string Id,double capacity,
    double Consumption,
    double Speed) : vehicleID(Id),AverageSpeed(Speed),EnergyConsumption(Consumption
        ),Capacity(capacity)
    {}
    string GetvehicleID()
    {
```

Output

```
Thanks For Visiting Our System
Total Bill Is 1100
_Locations Available For Delivery Are_
Clifton
Millinium
Saddar
Nazimabad
Gulshan
Malir
Enter Your Location For Order
Saddar
Time Of Delievery Is  = 3
The Optimum Delievery Route For Deilevery will Be Through  Gulshan
Active Delievers Are 4
Drone Taking The Route Where Moisture Lowest Is = 2 Location is = Saddar
Vehicle Id Of TimeShipYis Goining On A Visit In History
__Orders Available Are__ :
Biryani
Nihari
Haleem
Paya
Chapli Kebab
Sajji
Karahi
Aloo Paratha
Chana Chaat
Zarda
Do You Want To Place Order ? (Yes/No)
Yes
Enter Meal Name
Sajji
Do You Want To Place Order ? (Yes/No)
Yes
Enter Meal Name
Karahi
```

main.cpp

Share Run

```cpp
// 24k-0559 BAZIL-UDDIN-KHAN
#include <iostream>
using namespace std;

class Vehicle
{
    private:
    double Capacity;
    double EnergyConsumption;
    double AverageSpeed;



    static  int Activedelieveries;

    protected:
    string vehicleID;

    string Routes[6] = {"Clifton","Millinium","Saddar","Nazimabad","Gulshan","Malir"};
    double EstimatedTimeInHour[6] = {3.4,4,2,2,4,1.5};
    double Distance[6] ={2,5,3,7,8,9};
    string FoodAvailable[10] = {"Biryani", "Nihari", "Haleem", "Paya", "Chapli Kebab",
        "Sajji", "Karahi", "Aloo Paratha", "Chana Chaat", "Zarda"};
    double Prices[10] = {
        300.0, 350.0, 190.0, 350.0, 0.0, 1430.0, 960.0, 350.0, 490.0, 800};

    public:
    Vehicle(){vehicleID = " ";}

    Vehicle(string Id,double capacity,
    double Consumption,
    double Speed) : vehicleID(Id),AverageSpeed(Speed),EnergyConsumption(Consumption
        ),Capacity(capacity)
    {}
    string GetvehicleID()
    {
```

Output

```
Enter Meal Name
Sajji
Do You Want To Place Order ? (Yes/No)
Yes
Enter Meal Name
Karahi
Do You Want To Place Order ? (Yes/No)
No
Thanks For Visiting Our System
Total Bill Is 2390
_Locations Available For Delivery Are_
Clifton
Millinium
Saddar
Nazimabad
Gulshan
Malir
Enter Your Location For Order
Nz
Sorry We Donot Provide Seruve Here
_Locations Available For Delivery Are_
Clifton
Millinium
Saddar
Nazimabad
Gulshan
Malir
Enter Your Location For Order
Malir
Time Of Delievery Is  = 4.5
The Optimum Delievery Route For Deilevery will Be Through  Clifton
Active Delievers Are 5
From The Location Saddar And The Tunnel Has This Capaciry 47


=== Code Execution Successful ===
```

QUESTION4).

CODE:

```
//Bazil-Uddin-Khan 24k-0559

#include <iostream>
using namespace std;



int ProduceHash(string password)
{
   int hash = 5381;
   for(char c : password) {
      hash = hash * 33 + c;
   }
   return hash;
}


class User
{
```

```
    protected:
    string name;
    string id;
    string email;
    int hashed_password;
    int Hashed;
    string Studentpermission[1] = {"sumbit assignment"};
    string Tapermission[2] = {"view projects", "manage_students"};
    string Proffesorpermission[2] = {"assign
projects","full_lab_access"};
    string permissionlist;



    public:
    User()
    {
        name = " ";
        id = " ";
        email = " ";
        hashed_password =0;


    }
```

```
    User(string Name, string Id, string Email, string password,string list)
    : name(Name), id(Id), email(Email),permissionlist(list),
hashed_password(ProduceHash(password)) {}

    void displayinfo()
    {
        cout << " Name is " << name << endl;
        cout << " id is " << id << endl;
        cout << " Email is "<<email<< endl;
        cout << " Hashed password is " << hashed_password << endl;
        cout << " Permission list is " << permissionlist << endl;
    }

    bool authenticatepassword()
    {
        string password;
        cout << " Enter Password To Check Access: ";
        cin >> password;
        if (ProduceHash(password) == hashed_password)
        {
            cout << " Correct Password " << endl;
            return 1;
```

```
    }
    else
    {
        cout << " Wrong Password " << endl;
        return 0;
    }
}

void accesslab()
{
    if(permissionlist == "STUDENT")
    {


        string taskS;
        cout << " Welcome Student. Enter Your Task " << endl;
        cin.ignore();
        getline(cin,taskS);


        if(taskS == Studentpermission[0])
        {
```

```
        cout << " Access Granted Succesfully " << endl;
    }
    else
    {
        cout << " Invalid Entry. Access rejected  " << endl;

    }
}

else if(permissionlist == "TA")
{
    string taskT;
    cout << " Welcome Ta. Enter Your Task " << endl;
    cin.ignore();
    getline(cin,taskT);
     if(taskT == Tapermission[0] || taskT == Tapermission[1])
    {
        cout << " Access Granted Succesfully " << endl;
    }
    else
    {
        cout << " Invalid Entry. Access rejected  " << endl;
```

```
    }
  }

  else if(permissionlist == "PROFESSOR")
  {
    string taskP;
    cout << " Welcome Proffesor. Enter Your Task " << endl;
    cin.ignore();
    getline(cin,taskP);
    if(taskP == Proffesorpermission[0] || taskP ==
Proffesorpermission[1])
    {
      cout << " Access Granted Succesfully " << endl;
    }
    else
    {
      cout << " Invalid Entry. Access rejected  " << endl;

    }
  }
  else
```

```
    {
        cout << " Invalid Entry. Access rejected  " << endl;
    }

}

int Authentification(string Action)
{
    if(permissionlist == "STUDENT")
    {

     if(Action == "sumbit assignment")
        {
            cout << " Access Granted Succesfully " << endl;
            return 1;
        }
        else
        {
            cout << " Invalid Entry. Access rejected  " << endl;
            return 0;

        }
```

```
}

else if(permissionlist == "TA")
{

    if(Action == Tapermission[0] || Action == Tapermission[1])
   {
      cout << " Access Granted Succesfully " << endl;
      return 1;
   }
   else
   {
      cout << " Invalid Entry. Access rejected  " << endl;
      return 0;

   }
}

else if(permissionlist == "PROFESSOR")
{
```

```
        if(Action == Proffesorpermission[0] || Action ==
Proffesorpermission[1])
        {
           cout << " Access Granted Succesfully " << endl;
           return 1;
        }
        else
        {
           cout << " Invalid Entry. Access rejected  " << endl;
           return 0;

        }
      }
      else
      {
         cout << " Invalid Entry. Access rejected  " << endl;
         return 0;
      }

   }

};
```

```
int authenticateAndPerformAction(User * user, string action)
{
   bool result = user->authenticatepassword();
   if(result == 1)
   {
      cout  << " Succesfully Authentified Procced " << endl;
      int resalt = user->Authentification(action);
      if(resalt == 1)
      {
         cout << " Action Performed Successfully " << endl;
         return 1;
      }

      else
      {
         cout << " Sorry!.  Action didn't Performed Successfully " <<
endl;
         return 0;
      }
```

```
    }
    else
    {
        cout  << " Unsuccesfull. Authentication Failed. " << endl;
        return 0;

    }

}

class TA;
class Student : public User
{
    private:
    int Assignments[16] =
    {
        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    };

  static  int StudentSumbissionTracker;
    int date;
```

```
    int month;
    int year;

    public:
    Student() : date(0),month(0),year(0)
    {}

    Student(string Name, string Id,string Email,string charachter,string
list) : User( Name, Id,Email,charachter,list)
    {

    }

    void displayinfo(TA * ta);

    void GiveAssignment(string Assignmentname)
    {
        int dateOfSumbission;
        cout << " Enter Sumbission Date like from(1-31) " << endl;
        cin >> dateOfSumbission;
        if(dateOfSumbission > 0 && dateOfSumbission < 32)
        {
```

```
        date = dateOfSumbission;



        int Month;
        cout << " Enter Month (1-12) " << endl;
        cin >> Month;
        month = Month;

        if(Month > 0 && Month <13)
        {
           month = Month;
           int Year;
           cout << " Enter Year like 2025 etc " << endl;
           cin >> Year;
           year = Year;

        }

    }
}

void SubmitAssignment()
```

```
  {
    int Month, Date, Year;
    cout << " Enter Month Of Submission " << endl;
     cin >> Month;
     month = Month;

    cout << " Enter Date Of Submission " << endl;
    cin >> Date;
    date = Date;

   cout << " Enter Year Of Submission " << endl;
   cin >> Year;
   year = Year;

    if (Year > year || (Year == year && Month > month) || (Year ==
year && Month == month && Date > date))
    {
      cout << " Sorry. Submission deadline was " << date << "-" <<
month << "-" << year << endl;
      return;
     }

     if (StudentSumbissionTracker < 16)
```

```
    {
      Assignments[StudentSumbissionTracker] = 1;
      StudentSumbissionTracker++;
      cout << "Assignment submitted successfully!" << endl;
      cout << " S" << StudentSumbissionTracker << endl;
     }
     else
     {
      cout << " No more submissions allowed." << endl;
     }
    }


};
int Student :: StudentSumbissionTracker =0;

class TA : public Student
{
   private:
   string StudentList[20] =  {
   "Aiden", "Bella", "Caleb", "Daniel", "Elena", "Felix", "Grace",
"Henry", "Isla", "Jack",
```

```
   "Kaitlyn", "Liam", "Mia", "Nathan", "Olivia", "Paul", "Quinn",
"Ryan", "Sophia", "Thomas" };



   int ProjectCount[10];
   string TaProjectList[2];


   string TaStudents[16];


   int CurrentProjects;
   int Totaltastudents;
   string TaName;


  public:
  TA()
  {}


   TA(string Name, string Id,string Email,string charachter,string
list,string taname) : Student( Name, Id,Email,charachter,
list),TaName(taname)
  {
      CurrentProjects =0;
      Totaltastudents =0;
```

```
}

void AssignStudents()
{

    if(Totaltastudents > 16)
   {
      cout << " Cant Add " << endl;
      return;
    }

   string StudentName;
   cout << "Enter Student Name To Be Assigned To TA: ";
   cin >> StudentName;

   int found =-1;
   for(int i = 0; i < Totaltastudents; i++)
   {
     if(TaStudents[i] == StudentName)
      {
        cout << " Found " << endl;
        found =1;
```

```
    return;
  }
}


if(found ==-1)
{
  for(int i = 0; i < 20; i++)
  {
      if(StudentName == StudentList[i])
     {
     TaStudents[Totaltastudents] = StudentName;
       Totaltastudents++;


     }
   }
 }
 else
 {
 cout << " Student Already Assigned " << endl;
  }
```

```
}

void ViewCurrentProjects()
{
   for(int y =0; y< CurrentProjects ;y++)
   {
      cout << " Project " << y+1 << " is " << TaProjectList[y] << endl;
   }
}

void ProjectDetails()
{

   string Choice;
   cout << " Enter Choice In Yes/No if You want to start Project or not
" << endl;
   cin >> Choice;
   if(Choice == "Yes")
   {
      if (CurrentProjects >= 2)
```

```
        {
         cout << "TA can only work on 2 projects at a time." << endl;
        return;
        }
        string projectname;
        cout << " Enter Project name " << endl;
        cin >> projectname;

     TaProjectList[CurrentProjects] = projectname;
     cout << " Project Assigned Is " << projectname << endl;
         CurrentProjects++;
    }

 }

 void displayinfo()
{
  cout << "TA Students: " << endl;
  for(int y = 0; y < Totaltastudents; y++)
  {
     cout << "  " << TaStudents[y] << endl;
  }
```

```
 cout << "Projects Assigned to TA: " << endl;
 for(int y = 0; y < CurrentProjects; y++)
 {
    cout << "  Project " << y+1 << ": " << TaProjectList[y] << endl;
 }
}

int SearchTaStudent(string Studentname)
{
   for(int y =0; y< Totaltastudents;y++)
   {
      if(Studentname == TaStudents[y] )
      {
         return 1;
      }
   }
   return 0;
}

int GetProject()
{
```

```
    return CurrentProjects;
  }



};

void Student ::  displayinfo(TA * ta)

  {

    int result = ta->SearchTaStudent(name);
    if(result == 1)
    {
    cout << " Assignment Sumbission Status is " << endl;
    cout << name << " Sumbitted The Assignment " << endl;
    cout << " Email is " << email << endl;
    cout << " Id Is " << id << endl;
    for(int y =0; y < 16;y++)
    {
       cout << " Student " << y+1 << " Status is " << Assignments[y] <<
endl;
    }
```

```
    }
  }

class Proffessor : public User
{
  private:
  string ProffessorName;
  int Projects;

  public:
  Proffessor()
  {
    ProffessorName = " ";
  }

  Proffessor(string Name, string Id,string Email,string charachter,string lists,string name) : User( Name, Id, Email, charachter,lists)
  {
    ProffessorName = name;
  }

  void displayinfo()
```

```
    {
        cout << " Proffesor Name is " << ProffessorName << endl;
        cout << " Proffesor id is " << id << endl;
       cout << " Proffesor email is " << email << endl;

    }

    void WorkOnProject(TA *ta)
    {
       if (ta->GetProject() >= 2)
       {
        cout << "TA is already working on 2 projects." << endl;
        return;
       }

       ta->ProjectDetails();
       cout << "Professor assigned a project to the TA successfully!"  <<
endl;
     }



};
```

```
int main()
{

   cout << " _User Details_ " << endl;
   cout << endl;
   User user("Qasim","12k-8765","Qasim@nu.pk","1234567","TA");

   user.displayinfo();
   cout << endl;

   TA ta1("Alice", "T001", "alice@example.com", "tapass", "TA",
"Alice");

   cout << " Ta  Is Trying To Get Access to  Assigned Students for him
And Manage his projects " << endl;

   int Result = authenticateAndPerformAction(&ta1,
"manage_students");
   if(Result == 1)
   {
```

```
    ta1.AssignStudents();
    ta1.AssignStudents();
    ta1.AssignStudents();
    ta1.AssignStudents();
    cout << endl;
    cout << " _Ta Details_ " << endl;

    ta1.displayinfo();
    cout << endl;

    ta1.ProjectDetails();
    ta1.ProjectDetails();
    ta1.ProjectDetails();
    ta1.ViewCurrentProjects();



}
else
{
    cout << " Sorry Failed Authentification. Please Try Again " << endl;

}
```

```
   // My logic: The Constructor Name Should Of student should be in Ta Students assigned so that he can Sumbit Assignment
   Student s1("Henry", "S001", "john@example.com", "studentpass", "STUDENT");
   Student s2("Daniel","S002","Daniel@nu.edu.pk","studentpass2","STUDENT");


   cout << " Student Is Trying To Get Access to See Assigned Assignment And Sumbit It in time " << endl;


   int result = authenticateAndPerformAction(&s1, "sumbit assignment");
   if(result == 1)
   {
      s1.GiveAssignment("Mvc Assignment");
      s1.SubmitAssignment();
      cout << endl;
      cout << " _Student Details_ " << endl;
      cout << endl;
      s1.displayinfo(&ta1);// In Display MY Logic : Is That Student Can See only His Assignment Status so Others Status will be coming 0 so that  student cant see other Details
```

```
    }
    else
    {
        cout << " Sorry Failed Authentification. Please Try Again " << endl;


    }


    cout << " Student Is Trying To Get Access to See Assigned
Assignment And Sumbit It in time " << endl;
    int esult = authenticateAndPerformAction(&s2, "sumbit
assignment");
    if(esult == 1)
    {
        s2.GiveAssignment("Mvc Assignment");
        s2.SubmitAssignment();
        cout << endl;
        cout << " _Student Details_ " << endl;
        cout << endl;
        s2.displayinfo(&ta1);


    }
    else
```

```
    {
        cout << " Sorry Failed Authentification. Please Try Again " << endl;

    }



    Proffessor p1("Dr. Smith", "P001", "smith@example.com",
"professorpass", "PROFESSOR", "Dr. Smith");

    cout << " Proffesor Is Signing To Get Access to See Assigned
Assignment And GetAccess to lab" << endl;

    int REsult = authenticateAndPerformAction(&p1, "assign projects");
    if(REsult == 1)
    {
        p1.WorkOnProject(&ta1);
        cout << endl;
        cout << " _Proffesor Details_ " << endl;
        p1.displayinfo();
        cout << endl;

    }
```

```
    else
    {
        cout << " Sorry Failed Authentification. Please Try Again " << endl;


    }


    return 0;
}
```

OUTPUT Q4 :

main.cpp

```cpp
//Bazil-Uddin-Khan 24k-0559

#include <iostream>
using namespace std;



int ProduceHash(string password)
{
    int hash = 5381;
    for(char c : password) {
        hash = hash * 33 + c;
    }
    return hash;
}


class User
{

    protected:
    string name;
    string id;
    string email;
    int hashed_password;
    int Hashed;
    string Studentpermission[1] = {"sumbit assignment"};
    string Tapermission[2] = {"view projects", "manage_students"};
    string Proffesorpermission[2] = {"assign projects","full_lab_access"};
    string permissionlist;


    public:
    User()
    {
```

Output

```
_User Details_

Name is Qasim
id is 12k-8765
Email is Qasim@nu.pk
Hashed password is 939490257
Permission list is TA

Ta  Is Trying To Get Access to  Assigned Students for him  And Manage his projects
Enter Password To Check Access: tapass
Correct Password
Succesfully Authentified Procced
Access Granted Succesfully
Action Performed Successfully
Enter Student Name To Be Assigned To TA: Henry
Enter Student Name To Be Assigned To TA: Daniel
Enter Student Name To Be Assigned To TA: Sam
Enter Student Name To Be Assigned To TA: Bella

_Ta Details_
TA Students:
  Henry
  Daniel
  Bella
Projects Assigned to TA:

Enter Choice In Yes/No if You want to start Project or not
Yes
Enter Project name
C++
Project Assigned Is C++
Enter Choice In Yes/No if You want to start Project or not
No
Enter Choice In Yes/No if You want to start Project or not
No
Project 1 is C++
```

main.cpp

```cpp
//Bazil-Uddin-Khan 24k-0559

#include <iostream>
using namespace std;



int ProduceHash(string password)
{
    int hash = 5381;
    for(char c : password) {
        hash = hash * 33 + c;
    }
    return hash;
}


class User
{

    protected:
    string name;
    string id;
    string email;
    int hashed_password;
    int Hashed;
    string Studentpermission[1] = {"sumbit assignment"};
    string Tapermission[2] = {"view projects", "manage_students"};
    string Proffesorpermission[2] = {"assign projects","full_lab_access"};
    string permissionlist;


    public:
    User()
    {
```

Output

```
Project 1 is C++
Student Is Trying To Get Access to See Assigned Assignment And Sumbit It in time
Enter Password To Check Access: studentpass
Correct Password
Succesfully Authentified Procced
Access Granted Succesfully
Action Performed Successfully
Enter Sumbission Date like from(1-31)
8
Enter Month (1-12)
7
Enter Year like 2025 etc
2025
Enter Month Of Submission
1
Enter Date Of Submission
1
Enter Year Of Submission
2025
Assignment submitted successfully!
S1

_Student Details_

Assignment Sumbission Status is
Henry Sumbitted The Assignment
Email is john@example.com
Id Is S001
Student 1 Status is 1
Student 2 Status is 0
Student 3 Status is 0
Student 4 Status is 0
Student 5 Status is 0
Student 6 Status is 0
Student 7 Status is 0
Student 8 Status is 0
```

main.cpp

Share Run

```cpp
//Bazil-Uddin-Khan 24k-0559

#include <iostream>
using namespace std;



int ProduceHash(string password)
{
    int hash = 5381;
    for(char c : password) {
        hash = hash * 33 + c;
    }
    return hash;
}


class User
{

    protected:
    string name;
    string id;
    string email;
    int hashed_password;
    int Hashed;
    string Studentpermission[1] = {"sumbit assignment"};
    string Tapermission[2] = {"view projects", "manage_students"};
    string Proffesorpermission[2] = {"assign projects","full_lab_access"};
    string permissionlist;


    public:
    User()
    {
```

Output

```
Student 7 Status is 0
Student 8 Status is 0
Student 9 Status is 0
Student 10 Status is 0
Student 11 Status is 0
Student 12 Status is 0
Student 13 Status is 0
Student 14 Status is 0
Student 15 Status is 0
Student 16 Status is 0
Student Is Trying To Get Access to See Assigned Assignment And Sumbit It in time
Enter Password To Check Access: studentpass2
Correct Password
Succesfully Authentified Procced
Access Granted Succesfully
Action Performed Successfully
Enter Sumbission Date like from(1-31)
2
Enter Month (1-12)
2
Enter Year like 2025 etc
2025
Enter Month Of Submission
1
Enter Date Of Submission
1
Enter Year Of Submission
2025
Assignment submitted successfully!
S2

_Student Details_

Assignment Sumbission Status is
Daniel Sumbitted The Assignment
```

main.cpp

Share Run

```cpp
//Bazil-Uddin-Khan 24k-0559

#include <iostream>
using namespace std;



int ProduceHash(string password)
{
    int hash = 5381;
    for(char c : password) {
        hash = hash * 33 + c;
    }
    return hash;
}


class User
{

    protected:
    string name;
    string id;
    string email;
    int hashed_password;
    int Hashed;
    string Studentpermission[1] = {"sumbit assignment"};
    string Tapermission[2] = {"view projects", "manage_students"};
    string Proffesorpermission[2] = {"assign projects","full_lab_access"};
    string permissionlist;


    public:
    User()
    {
```

Output

```
Assignment Sumbission Status is
Daniel Sumbitted The Assignment
Email is Daniel@nu.edu.pk
Id Is S002
Student 1 Status is 0
Student 2 Status is 1
Student 3 Status is 0
Student 4 Status is 0
Student 5 Status is 0
Student 6 Status is 0
Student 7 Status is 0
Student 8 Status is 0
Student 9 Status is 0
Student 10 Status is 0
Student 11 Status is 0
Student 12 Status is 0
Student 13 Status is 0
Student 14 Status is 0
Student 15 Status is 0
Student 16 Status is 0
Proffesor Is Signing To Get Access to See Assigned Assignment And GetAccess to lab
Enter Password To Check Access: professorpass
Correct Password
Succesfully Authentified Procced
Access Granted Succesfully
Action Performed Successfully
Enter Choice In Yes/No if You want to start Project or not
Yes
Enter Project name
Clanguage
Project Assigned Is Clanguage
Professor assigned a project to the TA successfully!

_Proffesor Details_
Proffesor Name is Dr. Smith
Proffesor id is P001
```

main.cpp

Share | Run

```cpp
//Bazil-Uddin-Khan 24k-0559

#include <iostream>
using namespace std;



int ProduceHash(string password)
{
    int hash = 5381;
    for(char c : password) {
        hash = hash * 33 + c;
    }
    return hash;
}


class User
{

    protected:
    string name;
    string id;
    string email;
    int hashed_password;
    int Hashed;
    string Studentpermission[1] = {"sumbit assignment"};
    string Tapermission[2] = {"view projects", "manage_students"};
    string Proffesorpermission[2] = {"assign projects","full_lab_access"};
    string permissionlist;


    public:
    User()
    {
        name = " ";
```

Output

```
Student 2 Status is 1
Student 3 Status is 0
Student 4 Status is 0
Student 5 Status is 0
Student 6 Status is 0
Student 7 Status is 0
Student 8 Status is 0
Student 9 Status is 0
Student 10 Status is 0
Student 11 Status is 0
Student 12 Status is 0
Student 13 Status is 0
Student 14 Status is 0
Student 15 Status is 0
Student 16 Status is 0
Proffesor Is Signing To Get Access to See Assigned Assignment And GetAccess to lab
Enter Password To Check Access: professorpass
Correct Password
Succesfully Authentified Procced
Access Granted Succesfully
Action Performed Successfully
Enter Choice In Yes/No if You want to start Project or not
Yes
Enter Project name
Clanguage
Project Assigned Is Clanguage
Professor assigned a project to the TA successfully!

_Proffesor Details_
Proffesor Name is Dr. Smith
Proffesor id is P001
Proffesor email is smith@example.com



=== Code Execution Successful ===
```