Object oriented programing

Abdullah

21F-9248

Project : vaccine management system

Code:

```
#include<iostream>
#include<string>
#include<math.h>
#include<fstream>
#include<Windows.h>
using namespace std;
static int vacnited = 0;
static int Days_ = 1;
class Person
{
public:
```

```
        string firstName, lastName,
email;
        int id;
        long int contactNumber;
        Person()
        {
          firstName = "";
          lastName = "";
          string email = "";
          long int contactNumber =
0;
          string id = "";
        }
        // C++ program for the above
approach
        // Function to check the
character
        // is an alphabet or not
```

```
        bool isChar(char c)
        {
          return ((c >= 'a' && c <= 'z')
                || (c >= 'A' && c <= 'Z'));
        }
        // Function to check the
character
        // is an digit or not
        bool isDigit(const char c)
        {
          return (c >= '0' && c <= '9');
        }
        // Function to check email id
is
        // valid or not
        bool is_valid(string email)
        {
```

```
        // Check the first character
        // is an alphabet or not
        if (!isChar(email[0])) {

            // If it's not an alphabet
            // email id is not valid
            return 0;
        }
        // Variable to store position
        // of At and Dot
        int At = -1, Dot = -1;

        // Traverse over the email id
        // string to find position of
        // Dot and At
        for (int i = 0; i <
email.length(); i++) {
```

```
        // If the character is '@'
        if (email[i] == '@') {

            At = i;
        }

        // If character is '.'
        else if (email[i] == '.') {

            Dot = i;
        }
    }

    // If At or Dot is not present
    if (At == -1 || Dot == -1)
        return 0;
```

```
            // If Dot is present before At
            if (At > Dot)
                return 0;

            // If Dot is present at the
end
            return !(Dot >=
(email.length() - 1));
        }
        void Record()
        {
          cout << "Please Enter ID: ";
          cin.ignore();
          cin >> id;
```

```
        cout << "Please Enter your
First Name: ";
        cin >> firstName;
        //      cin.ignore();
        //      getline(cin, firstName);
        cout << "Please Enter your
Last Name: ";
        cin >> lastName;
        //      cin.ignore();
        //      getline(cin,lastName);
        cout << "Please Enter your
Email: ";
        cin >> email;

        bool v = is_valid(email);
        if (v) {
            cout << email << " : "
                << "valid" << endl;
```

```
            }
            else {
                cout << email << " : " <<
"Invalid" << endl;
                cout << "Please Enter
your Email: ";
                cin >> email;
                is_valid(email);
            }
            cout << "Please Enter your
Contact Number: ";
            cin >> contactNumber; //
need to make a function for contact
number verification
        }
        void display()
        {
```

```
            cout << "First name " <<
firstName << endl;
            cout << "Last name " <<
lastName << endl;
            cout << "Email " << email
<< endl;
            cout << "Id " << id << endl;
            cout << "Contact Number "
<< contactNumber << endl;
        }
};
class SuperAdmin :public Person
{
public:
        int Insert(Person* p, int n)
        {
          Record();
        }
```

```
        void Search(Person* p, int
ID, int n)
        {
          int i = 0;
          for (i = 0; i < n; i++)
          {
              if (p[i].id == ID)
              {
                  cout <<
"\tNAME\id\temail\ncontactNumber\
n==========================
====================\n";
                  cout << id << "\t" <<
firstName << "\t" << email << "\n" <<
contactNumber << "\n";
                  break;
              }
          }
```

```
            if (p[i].id != ID)
            {
                cout << "\nRECORD
NOT FOUND.\n";
            }
        }
        int Del(Person* p, int n, int
ID)
        {
            int j = 0, k, flag = 0;
            for (j = 0; j < n; j++)
            {
                if (p[j].id == ID)
                {
                    flag = 1;
                    break;
                }
```

```
        }
        if (flag == 1)
        {
            for (k = j; k < n; k++)
            {
                p[k] = p[k + 1];
            }
            cout << "\nRECORD
DELETED.\n";
            return n - 1;
        }
        else
        {
            cout << "\nRecord Not
Found\n";
            return n;
        }
```

```
        }
        int Update(Person* p, int ID,
int n)
        {
          int i, ch1;
          for (i = 0; i < n; i++)
          {
              if (p[i].id == ID)
              {
                  while (1) {
                      cout <<
"\n!!===OPTIONS IN
UPDATE===!!\n";
                      cout << "\n 1.
Update firstname";
                      cout << "\n 2.
Update email";
```

```
                        cout << "\n 3.
Update Both";
                        cout << "\n 4.
Return to main Menu";
                        cout << "\n\n Enter
Your Choice:";
                        cin >> ch1;
                        switch (ch1) {
                        case 1: cout <<
"First Name:";
                            cin >>
p[i].firstName;
                            cout <<
"Record Updated...\n";
                            break;
                        case 2: cout <<
"email:\t";
                            cin >>
p[i].email;
```

```
                                cout <<
"Record Updated...\n";
                                break;
                            case 3: cout <<
"first name:";
                                cin >>
p[i].firstName;
                                cout <<
"Email:\t";
                                cin >>
p[i].email;
                                cout <<
"Record Updated...\n";
                                break;
                            case 4: return n;
                            default: cout << "!!
Wrong Key !!";
                                break;
```

```
                    }
                }
                break;
            }
        }
        if (p[i].id != ID)
        {
            cout << "\nRecord Not Found\n\n";
        }
    }
};
class Admin :public Person
{
    int days;
    long int population = 200000000;
```

```
        friend class Warehouse;
        int vaccines =
50000000000000;
        friend class
GovernmentOffical;
public:
        void
SetupVaccinationCentres() {
          int* vaccineCenter;
          vaccineCenter = new
int[population / 20000];
          int* city = new
int[*vaccineCenter];

        }
        void DeleteVaccines() {
          ++days;
```

```
            cout << "Enter
consumption of Vaccination
Center";
            int* VaccineforCenter =
new int;
            cin >> *VaccineforCenter;
            int newvaccines =
*VaccineforCenter;
            cout << "You need to
provide this many vaccines at the
earliest:" << newvaccines;
            vaccines -=
*VaccineforCenter;


        }
};
class Warehouse {
            int vaccines =
50000000000000;
```

```
        int warehouses =
50000000000000 / 20000;
        int vaccinePerWarehouse =
20000;
public:
        Warehouse() {}
        void DeleteWarehouse() {
          Admin a1;
          a1.DeleteVaccines();
        }
};
class FrontDeskOfficer : public
Person
{


};
class GovernmentOffical :public
Person
```

```
{
        Admin object;
        double percentage;
        long int Money;
public:
        void calculation()
        {
          long int temp = vacnited;
          percentage = (temp /
object.population) * 100;
          cout << "This is the
percentage of vaccnited people :\n"
<< percentage << endl;
        }
        void amount_cal()
        {
          int money = 0;
```

```
            cout << "Enter Amount of 1
vaccine :";
            cin >> money;
            Money = money *
object.vaccines;
            cout << "Total money
spend on vaccines :" << Money <<
endl;
        }
        void display_everydata()
        {
          string out;
          fstream allData;
          allData.open("AllData.txt",
ios::in);
          while (!allData.eof())
          {
              getline(allData, out);
```

```
                if (out != "- 1")
                {
                    cout << out;
                }
                if (out == "-")
                {
                    cout << "Had a
Booster shot ";
                }
                if (out == "_")
                {
                    cout << "Fully
vaccinated";
                }
                if (out == "false")
                {
                    cout << "Half
Vaccined";
```

```
            }
            cout << endl;
        }
    }
};
class Citizen :public Person
{
private:
        string firstName, lastName, phoneNumber,
          email, password, cnic, bloodType, city, state, address, dob, username;
public:
        Citizen()
        {
          string firstName = " ";
          string lastName = " ";
```

```
        string phoneNumber = " ";
        string email = " ";
        string cnic = " ";
        string bloodType = " ";
        string city = " ";
        string state = " ";
        string address = " ";
        string dob = " ";
        string passward = " ";
    }
    Citizen(string firstname,
string lastname, string
phonenumber, string email, string
password, string cnic, string
bloodtype, string city, string state,
string address, string dob)
    {
```

```
            this->firstName =
firstname;
            this->lastName = lastname;
            this->phoneNumber =
phonenumber;
            this->email = email;
            this->password =
password;
            this->cnic = cnic;
            this->bloodType =
bloodtype;
            this->city = city;
            this->state = state;
            this->address = address;
            this->dob = dob;
            fstream allData;
            allData.open("AllData.txt",
ios::out | ios::app);
```

```
            allData << " First name " <<
firstName << "\nLast Name " <<
lastName << "\nPhone number " <<
phoneNumber << "\nEmail " <<
email << "\nPassword " << this-
>password << "\nCNIC " << this-
>cnic << endl;
            allData << "Blood Type " <<
this->bloodType << "\nCity " << this-
>city << "\nState " << this->state <<
"\nAddress " << this->address <<
"\nDOB " << this->dob << endl;
             allData.close();
           }
           void Initlzation()
           {
             fstream sign_up;
```

```
sign_up.open("Sign_up_data.txt", ios::out | ios::app);
            fstream Login;

          Login.open("login_data.txt", ios::out | ios::app);
            Login << endl;
            cout << "CNIC :";
            cin >> cnic;
            sign_up << "CNIC " << cnic << endl;
            cin.ignore();
            cout << "Enter First name :";
            getline(cin, firstName);
            sign_up << "First name " << firstName << endl;
            Login << firstName;
```

```
            cout << "Enter Last name
:";
            getline(cin, lastName);
            sign_up << "Last name " <<
lastName << endl;
            Login << lastName << endl;
            cout << "Enter Phone
number :";
            cin >> phoneNumber;
            sign_up << "Phone number
" << phoneNumber << endl;
            cout << "Enter Email ";
            cin >> email;
            sign_up << "Email " <<
email << endl;
            cout << "Enter City ";
            cin >> city;
```

```
            sign_up << "City " << city
<< endl;
            cout << "Enter address ";
            getline(cin, address);
            sign_up << "Address " <<
address << endl;
            cout << "Enter state ";
            getline(cin, state);
            sign_up << "State " <<
state << endl;
            cout << "Enter blood Type
";
            cin >> bloodType;
            sign_up << "Blood type "
<< bloodType << endl;
            cout << "Enter password ";
            cin >> password;
            Login << password;
```

```
            cout << "Enter DOB ";
            cin >> dob;
            sign_up << "DOB " << dob
<< endl << "-1" << endl;
        }
        void signup()
        {
          int ec = 0;
          int age;
          cout << " the firstName  ";
          cout << firstName;
          cout << endl;
          cout << "Enter the
lastName  ";
          cout << lastName;
          cout << endl;
          cout << " the phoneNumbe
";
```

```
            cout << phoneNumber;
            cout << endl;
            cout << " the email,   ";
            cout << email << endl;
            cout << "Enter the
password  ";
            cin >> password;
            cout << endl;
            cout << " the cnic  ";
            cout << cnic;
            cout << endl;
            cout << " the bloodType  ";
            cout << bloodType;
            cout << endl;
            cout << " the city ";
            cout << city;
            cout << endl;
```

```
            cout << " the  state  ";
            cout << state;
            cout << endl;
            cout << " the address   ";
            cout << address;
            cout << endl;
            cout << " the   dob  ";
            cout << dob;
            cout << endl;
        }
        void history()
        {
            string name, cnic, dno, vn;
            /*
                array - data store: name
, cnic , vaccine dosage number ,
vaccine name
```

```
            */


        }
        void login()
        {
            cout << "Enter Login
Details :\n";
            cout << "Enter user name
:";
            cin >> username;
            cout << "Enter password
:";
            cin >> password;
            fstream Login;
            string un;
            string pw;
            bool Login_check = false;
```

```
            Login.open("login_Data.txt",
ios::in);
             while (!Login.eof())
             {
                 if (Login_check == false)
                 {
                     getline(Login, un);
                 }
                 if (un == username)
                 {
                     Login_check = true;
                     getline(Login, pw);
                     if (pw == password)
                     {
                         break;
                     }
```

```
            }
        }
        if (un != username || pw !=
password) {


SetConsoleTextAttribute(GetStdHan
dle(STD_OUTPUT_HANDLE), 4);
            cout << "\aInvalid User
name or Password \n";


SetConsoleTextAttribute(GetStdHan
dle(STD_OUTPUT_HANDLE), 7);
            login();
        }
        Login.close();
        cout << "Logged in
Sucessfully\n";
```

```
            int age = 0, ec = 0;
            cout << "Enter Your Age :
";
            cin >> age;
            if (age >= 5)
            {
                cout << "Choose:\n1.
Not allergic to Component of
Vaccine\n2. Allergic to Components
of Vaccine\nI'll choose : ";
                cin >> ec;
                if (ec == 1)
                {
                    cout << "You may
proceed for vaccination\n";
                }
                else if (ec == 2)
                {
```

```
                    cout << "Sorry, You
can not proceed for vaccination\n";
                }
            }
            else
            {
                cout << "User not
registered\n";
                signup();
                cout << "Enter the
firstName  ";
                cin >> firstName;
                cout << "Enter the
lastName  ";
                cin >> lastName;
                cout << "Enter the
phoneNumbe   ";
                cin >> phoneNumber;
```

```
                cout << "Enter the email,
";
                cin >> email;
                cout << "Enter the
password  ";
                cin >> password;
                cout << "Enter the cnic
";
                cin >> cnic;
                cout << "Enter the
bloodType  ";
                cin >> bloodType;
                cout << "Enter the   city
";
                cin >> city;
                cout << "Enter the  state
";
                cin >> state;
```

```
                cout << "Enter the
address   ";
                cin >> address;
                cout << "Enter the   dob
";
                cin >> dob;
                Citizen cn(firstName,
lastName, phoneNumber,
                    email, password,
cnic, bloodType, city, state,
address, dob);
                cn.login();
              }
            }
            void Second_Booster()
            {
              int ch;
              string temp;
```

```
            int check1 = 0, check2 = 0;
            cout << "So this is 1- Your
second dose\n OR \n2- you want
booster dose\n";
            cin >> ch;
            if (ch == 1)
            {
                int abc;
                fstream allData;
                string rep = "false";
                string repl1;

           allData.open("AllData.txt",
ios::in | ios::out | ios::app);
                getline(allData, temp);
                if (temp == "{" || check1
== 1)
                {
```

```
                    if (check1 == 1)
                    {
                        check1 == 0;
                        abc = stoi(temp);
                    }
                    if (check1 == 0) {
check1++; }
                }
                if (temp == "false") {
                    if (Days_ == abc) {

            repl1.replace(repl1.find(rep),
rep.length(), "_");
                    }
                }
            }
            else if (ch == 2)
            {
```

```
                int abc;
                fstream allData;
                string rep = "false";
                string rep1 = "true";
                string repl1;

            allData.open("AllData.txt",
ios::in | ios::out | ios::app);
                getline(allData, temp);
                if (temp == "{" || check1
== 1)
                {
                    if (check1 == 1)
                    {
                        check1 == 0;
                        abc = stoi(temp);
                    }
```

```
                    if (check1 == 0) {
check1++; }
                }
                if (temp == "false" ||
temp == "true") {
                    if (Days_ == abc) {

            repl1.replace(repl1.find(rep),
rep.length(), "-");


repl1.replace(repl1.find(rep1),
rep1.length(), "-");
                    }
                }
                allData.close();
             }
           }
```

```
        void doctor()
        {
          int blood_pressure,
oxygen_level, glucose_level;
          cout << "Dr checked the
blood_pressure :" << endl;
          cin >> blood_pressure;
          cout << "Dr checked the
oxygen_level :" << endl;
          cin >> oxygen_level;
          cout << "Dr checked the
glucose_level :" << endl;
          cin >> glucose_level;
          if (glucose_level < 120 &&
blood_pressure < 125 &&
oxygen_level < 100)
          {
```

```
                cout << "u can move for
vaccine \n ";


            }
            else {
                cout << "wait for the
stable of your glucose_level,
blood_pressure, oxygen_level  " <<
endl;
            }
          }
          void Get_vaccine()
          {
            vacnited++;
            int choose;
            cout << " citizen wanted
which vaccine \n IF pizer press 1 \n
```

```
IF sanovac press 2 \n IF sanoform
press 3 \n ";
            cin >> choose;
            fstream allData;
            allData.open("AllData.txt",
ios::out | ios::app);
            if (choose == 1)
            {
                cout << "you get first
dose of VACCINE PIZER " << endl;
                allData << "you get first
dose of VACCINE PIZER " << endl;
            }
            if (choose == 2)
            {
                cout << "you get first
dose of VACCINE SANOVAC " <<
endl;
```

```
                allData << "you get first
dose of VACCINE SANOVAC " <<
endl;
            }
            if (choose == 3)
            {
                cout << "you get first
dose of VACCINE SANOFORM " <<
endl;
                allData << "you get first
dose of VACCINE SANOFORM " <<
endl;
            }
            int date, month, year, time;
            cout << "Entered the date
of  geting dose of vaccine " << endl;
            cin >> date;
            cout << "Entered the mouth
of  geting dose of vaccine " << endl;
```

```
            cin >> month;
            cout << "Entered the year
of  geting dose of vaccine " << endl;
            cin >> year;
            cout << "Entered the Time
of  geting dose of vaccine " << endl;
            cin >> time;
            allData << "Vaccine date "
<< date << "/" << month << "/" <<
year << endl;
            cout << "THANKS FOR
GETTING VACCINE FOR SAFETY
OF COUNTRTY " << endl;
            if (Days_ == 31)
            {
                Days_ = 1;
            }
            Days_++;
```

```
            if (month < 12)
            {
                cout << "Your next
Vaccine will be " << date << " month
" << month + 1 << endl;
                allData << "Your next
Vaccine will be " << date << " month
" << month + 1 << endl;
            }
            else if (month == 12)
            {
                month = 1;
                cout << "Your next
Vaccine will be " << date << " month
" << month << " year " << year <<
endl;
                allData << "{\nYour next
Vaccine will be " << date << " month
```

```
" << month << " year " << year <<
endl << "false\n-1";
                }
                allData.close();
            }
};
int main()
{
            string choice;
            Citizen obj;
            GovernmentOffical abc;
            Person* object;
            //SuperAdmin object_of;
            //object_of.Insert(object);
line:

SetConsoleTextAttribute(GetStdHan
dle(STD_OUTPUT_HANDLE), 4);
```

```
            cout << "\t\t\tWelcome to
Vaccination Center ";

SetConsoleTextAttribute(GetStdHan
dle(STD_OUTPUT_HANDLE), 7);
            cout << "Select";

SetConsoleTextAttribute(GetStdHan
dle(STD_OUTPUT_HANDLE), 5);
            cout << "Sign Up";

SetConsoleTextAttribute(GetStdHan
dle(STD_OUTPUT_HANDLE), 7);
            cout << "OR";

SetConsoleTextAttribute(GetStdHan
dle(STD_OUTPUT_HANDLE), 3);
            cout << "Login \n";
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
            getline(cin, choice);
            if (choice != "sign up" && choice != "login")
            {
              goto line;
            }
            while (1)
            {
              if (choice == "sign up")
              {
                  obj.Initlzation();
              }
              if (choice == "login")
              {
                  obj.login();
```

```
            }
            obj.doctor();
            obj.Get_vaccine();
            cout << "Do you want
booster or ur next vaccine :";
            cin >> choice;
            if (choice == "booster" ||
choice == "vaccine")
            {
                obj.Second_Booster();
            }
            abc.calculation();
            abc.amount_cal();
            Sleep(1000);
            abc.display_everydata();
            Sleep(5000);
            system("cls");
```

```
            goto line;
        }
        system("pause");
}
```