**Ganpat University**

**Faculty of Engineering & Technology**

**Computer Science & Engineering**

***Name:- Dwij Vatsal Desai***

***Sem:- 2***

***Sub: - ESFP-II***

***Enrollment No.:- 23162121027***

***Prac:- 9***

***Date:- 16/4/2024***

**Practical 9**

***Definition:***

1. First create one login module, where you have to enter login id and password,

if given id and password is correct then main program functionality should be

open. On incorrect login, you have to give maximum three chance for login,

otherwise program should be terminated.

2. Minimum 1 constructor method should be available in the program, rest as per

your program functionality requirement. You can also create/declare user

defined function as per your requirement.

3. For the solving purpose of given topic practical, you need to create minimum

three classes, rest as per your given object information requirement.

4. Implement the concept of friend class for the performing purpose of the

practical. Second class should be the friend class of the first class. Friend class

should be declared under private access specifier in the first class.

5. Second class should have minimum one friend function, rest you can declare

as per your practical program functionality requirement.

6. Third class should be the friend class of second class, and accordingly you

have to perform practical.

7. You must use access specifier for data member and member function

declaration in program.

8. Wherever is required to use character data member in class, instead of that use

compulsorily string data member.

9. Take minimum 10 data/record from the user and display according to the choice

of user category wise. (Minimum six different options should be there for

displaying information, and if you want more as per program requirement you

can add more choices).

10.Use all possibility filter method from stored record information.

11.After all functionality execution, you need to call destructor function.

***Code:-***

#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

#include <unordered\_set>

using **namespace** std;

*// Forward declaration*

**class** elearning;

*// Friend function declaration*

**void** storeDataToFile(elearning**&** obj);

**class** elearning

{

**private:**

*// choices made in course selection (for data storing in file handling)*

**int** expression;

    string selectedCourse;

    string courseLevel;

    string preferredWayOfLearning;

    string durationForEachDay;

*// User info*

    string age;

    string name;

    string address;

    string email;

    string tinfo = "";

*// Set of valid usernames*

    unordered\_set<string> validUsernames = {"DIJ"};

**public:**

*// Default constructor*

    elearning()

    {

        expression = 0; *// Default value*

    }

*// Parameterized constructor*

    elearning(**int** exp) *//Course choices*

    {

        expression = exp;

        switch (expression)

        {

        case 1:

            cout << endl << "You chose Business" << endl << endl;

            selectedCourse = "Business";

            selectComputerCourse();

            break;

        case 2:

            cout << endl << "You chose Computer course" << endl << endl;

            selectedCourse = "Computer course";

            selectComputerCourse();

            break;

        case 3:

            cout << endl << "You chose Data science" << endl << endl;

            selectedCourse = "Data science";

            selectComputerCourse();

            break;

        case 4:

            cout << endl << "You chose ICT" << endl << endl;

            selectedCourse = "ICT";

            selectComputerCourse();

            break;

        default:

            break;

        }

    }

**void** selectCourse()

    {

        cout << ">>>Course Selection<<<" << endl;

        cout << " Business" << endl;

        cout << " Computer course" << endl;

        cout << " Data science" << endl;

        cout << " ICT" << endl;

    }

**void** selectComputerCourse()

    {

**int** totalDuration = 0;

**int** innerExpression = 0;

        while (true)

        {

            cout << endl << ">>>Enter choice for Computer course:<<<" << endl;

            cout << " Course Level" << endl;

            cout << " Preferred Way of Learning" << endl;

            cout << " Duration for Each Day" << endl;

            cout << " Exit" << endl;

            cin >> innerExpression;

            switch (innerExpression)

            {

            case 1:

            {

**int** level = 0;

                cout << "Course Level:" << endl;

                cout << " 1-2 weeks (basic)" << endl;

                cout << " 2-5 weeks (intermediate)" << endl;

                cout << " 6-7 weeks (advance)" << endl;

                cin >> level;

                switch (level)

                {

                case 1:

                    cout << "1-2 weeks (basic)" << endl;

                    courseLevel = "1-2 weeks (basic)";

                    totalDuration += 2; *// Add 2 weeks to total duration*

                    break;

                case 2:

                    cout << "2-5 weeks (intermediate)" << endl;

                    courseLevel = "2-5 weeks (intermediate)";

                    totalDuration += 5; *// Add 5 weeks to total duration*

                    break;

                case 3:

                    cout << "6-7 weeks (advance)" << endl;

                    courseLevel = "6-7 weeks (advance)";

                    totalDuration += 7; *// Add 7 weeks to total duration*

                    break;

                default:

                    cout << "Invalid choice" << endl;

                    break;

                }

                break;

            }

            case 2:

            {

**int** preferredWay = 0;

                cout << "Preferred Way of Learning:" << endl;

                cout << " Guided project" << endl;

                cout << " Personal practical" << endl;

                cout << " Notes and concept learning" << endl;

                cin >> preferredWay;

                switch (preferredWay)

                {

                case 1:

                    cout << "Guided project" << endl;

                    preferredWayOfLearning = "Guided project";

                    break;

                case 2:

                    cout << "Personal practical" << endl;

                    preferredWayOfLearning = "Personal practical";

                    break;

                case 3:

                    cout << "Notes and concept learning" << endl;

                    preferredWayOfLearning = "Notes and concept learning";

                    break;

                default:

                    cout << "Invalid choice" << endl;

                    break;

                }

                break;

            }

            case 3:

            {

**int** duration = 0;

                cout << "Duration for Each Day:" << endl;

                cout << " 2 hours" << endl;

                cout << " 4 hours" << endl;

                cout << " 6 hours" << endl;

                cin >> duration;

                switch (duration)

                {

                case 1:

                    cout << "2 hours" << endl;

                    durationForEachDay = "2 hours";

                    break;

                case 2:

                    cout << "4 hours" << endl;

                    durationForEachDay = "4 hours";

                    break;

                case 3:

                    cout << "6 hours" << endl;

                    durationForEachDay = "6 hours";

                    break;

                default:

                    cout << "Invalid choice" << endl;

                    break;

                }

                break;

            }

            case 4:

                cout << "Exiting..." << endl;

                cout << "Total duration to complete the course: " << totalDuration << " weeks" << endl;

                storeDataToFile(\*this); *// Call friend function with current object*

                return;

            default:

                cout << "Invalid choice" << endl;

                break;

            }

        }

    }

*// Friend function declaration*

**friend** **void** storeDataToFile(elearning**&** obj);

*// Friend function definition*

**friend** **void** userinfo(elearning**&** obj);

**friend** **void** signpage(elearning**&** obj3);

**friend** **void** loginpage(elearning**&** obj2);

};

**void** signpage(elearning**&** oj)

{

}

**void** loginpage(elearning**&** obj2)

{

    cout << "Enter your username: ";

    string username;

    cin >> username;

*// Convert entered username to uppercase for case-insensitive comparison*

    transform(username.begin(), username.end(), username.begin(), ::toupper);

*// Check if username exists in the set of valid usernames*

    if (obj2.validUsernames.find(username) == obj2.validUsernames.end())

    {

        cout << "User does not exist." << endl;

        return;

    }

    cout << "Enter your password: ";

    string password;

    cin >> password;

*// Perform authentication logic here*

*// For simplicity, we will just print the username and password*

    cout << "Username: " << username << endl;

    cout << "Password: " << password << endl;

}

**void** storeDataToFile(elearning**&** obj)

{

    ofstream file("elearning\_data.txt", ios\_base::app);

    if (!file.is\_open())

    {

        cout << "Error opening file" << endl;

        return;

    }

    file << "Total user data:-\nAge  Name  Address  Email" << endl;

    file << obj.tinfo << endl

         << endl;

    file << "Course Info:" << endl;

    file << "Selected Course: " << obj.selectedCourse << endl;

    file << "Course Level: " << obj.courseLevel << endl;

    file << "Preferred Way of Learning: " << obj.preferredWayOfLearning << endl;

    file << "Duration for Each Day: " << obj.durationForEachDay << endl;

    file << "---------------------------------------" << endl;

    file.close();

    cout << "Data saved to elearning\_data.txt" << endl;

}

**void** userinfo(elearning**&** obj) *//string management (Practical\_8)*

{

    cout << "Enter your info here:-(Age,Name,Address,Email)" << endl;

    cin >> obj.age >> obj.name >> obj.address >> obj.email;

    cout << endl;

*// Append()*

    obj.tinfo = obj.age + " " + obj.name + " " + obj.address + " " + obj.email;

*//empty()*

    if (obj.age.empty() || obj.name.empty() || obj.address.empty() || obj.email.empty())

    {

      cout << "Error: Data not stored\n\n";

      cout << "Enter your info again here:-(Age,Name,Address,Email)" << endl;

      userinfo(obj);

    }

*// Insert()*

    string city = "City: ";

    obj.address.insert(0, city);

*// At*

**char** firstChar = obj.address.at(0); *//to order it in alphabetical-order*

*// Length*

**int** addressLength = obj.address.length();

*// Size*

**int** emailSize = obj.email.size();

*// Clear*

    obj.age.clear();

*// Erase*

    obj.address.erase(0, 6); *// Erase "City: " from address*

*// Empty*

**bool** isNameEmpty = obj.name.empty();

*// Sort (sorting the name in alphabetical order)*

    sort(obj.name.begin(), obj.name.end());

*// To Upper*

    transform(obj.name.begin(), obj.name.end(), obj.name.begin(), ::toupper); *// Convert name to uppercase*

*// Substring*

    string userName = obj.name.substr(0, 3); *// Get first 3 characters for username*

    cout << "Your user name will be: " << userName << endl;

}

**int** main()

{

*//<<< This is user interface >>>*

**int** Uchoice;

*// Declare elearning objects outside the switch*

    elearning userObj;

    elearning courseObj;

    elearning loginObj;

    cout << "Welcome to the E-learning System" << endl;

    while (true)

    {

        cout << endl << "<<< Main Menu >>>" << endl;

        cout << "1. Sign Up" << endl;

        cout << "2. Login Page" << endl;

        cout << "3. Course Selection" << endl;

        cout << "4. Exit" << endl;

        cout << "Enter your choice: ";

        cin >> Uchoice;

        cin.ignore(); *// Remove newline character from buffer*

        switch (Uchoice)

        {

        case 1:

            userinfo(userObj); *// Call userinfo function*

        break;

        case 2:

            loginpage(loginObj); *// Call loginpage function*

        break;

        case 3:

**int** userChoice;

            cout << "Enter your choice: ";

            cin >> userChoice;

            cin.ignore(); *// Remove newline character from buffer*

*// Create courseObj with the user's choice*

            courseObj = elearning(userChoice);

            courseObj.selectCourse(); *// Call selectCourse function*

        break;

        case 4:

            cout << "Exiting..." << endl;

        return 0;

        default:

            cout << "Invalid choice" << endl;

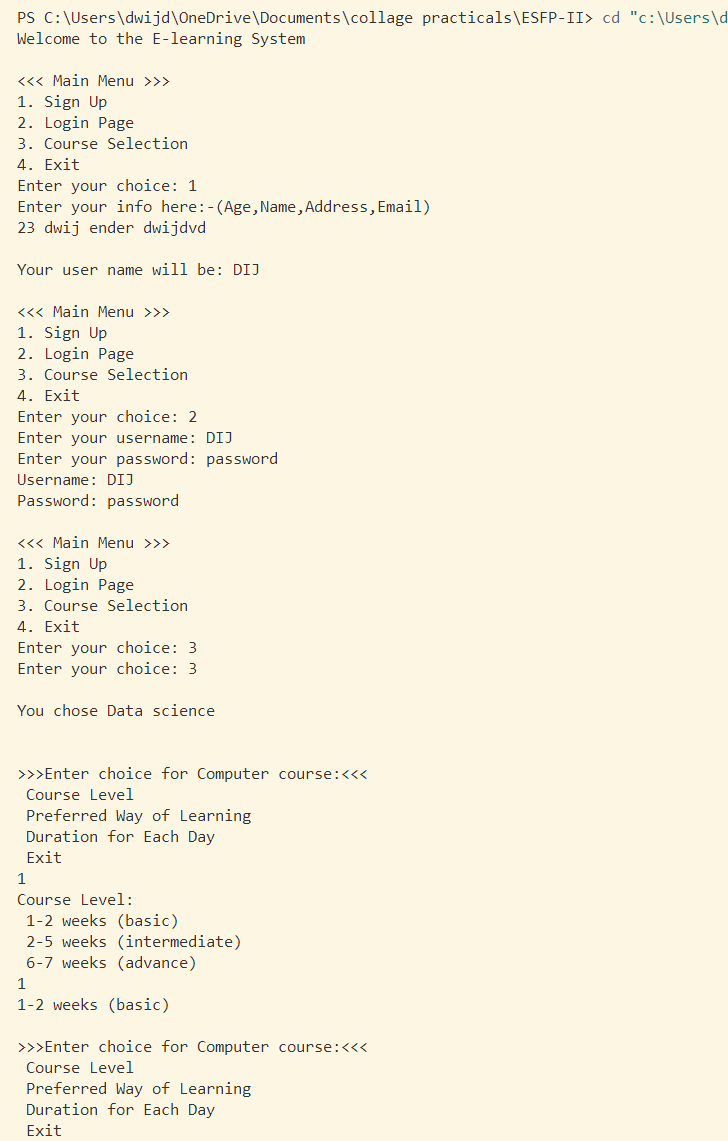
        break;

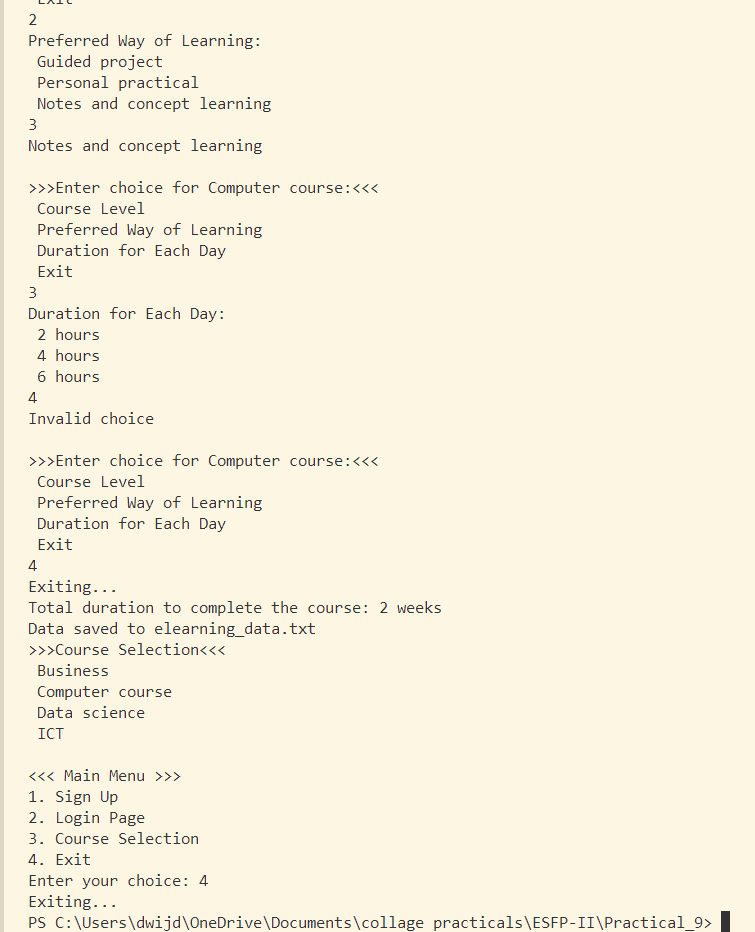
        }

    }

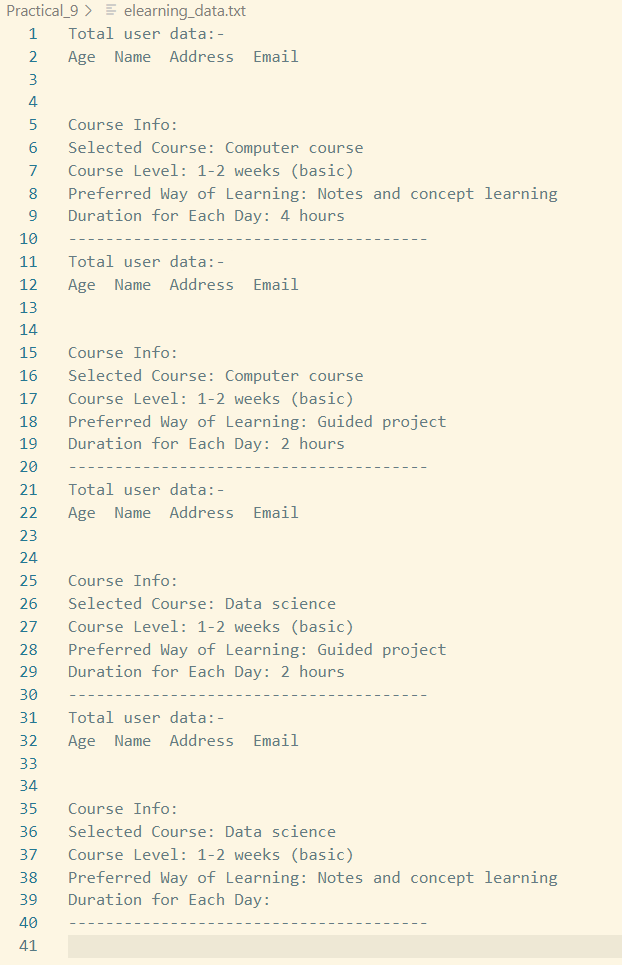
    return 0;

}

***Output:-***

******

***Recorded data in text file:-***



***Photo:-*** 