DSA Lab

Mr. ALEEM AHMAD

Bahria University

Open Ended 1

LAB Journal

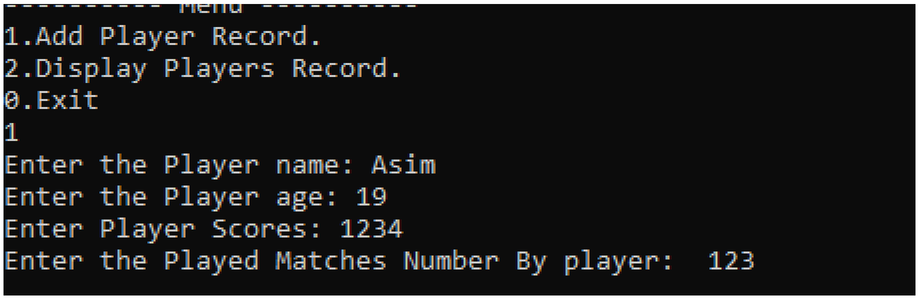
Asim Ali (01-131232-015)

Open Ended 1

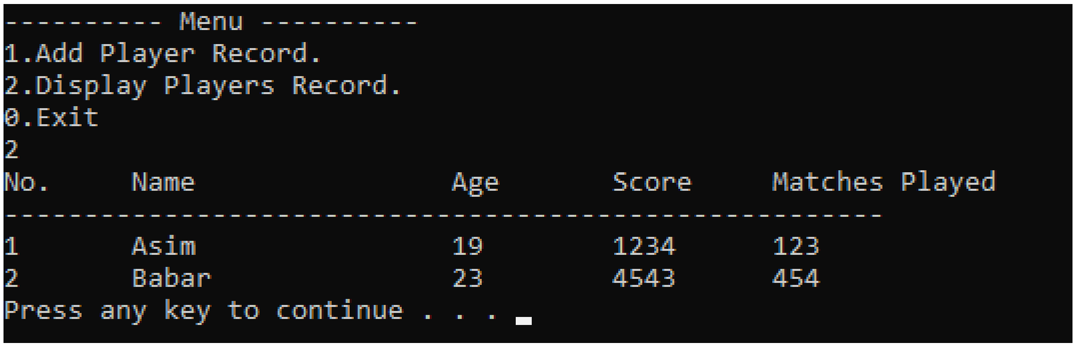
Lab Task GitHub Link:

Link

OUTPUT:



```
---------- Menu ----------
1.Add Player Record.
2.Display Players Record.
0.Exit
1
Enter the Player name: Asim
Enter the Player age: 19
Enter Player Scores: 1234
Enter the Played Matches Number By player:  123
```



```
---------- Menu ----------
1.Add Player Record.
2.Display Players Record.
0.Exit
2
No.     Name                    Age         Score       Matches Played
-------------------------------------------------------
1       Asim                    19          1234        123
2       Babar                   23          4543        454
Press any key to continue . . .
```

CODE:

```
#include<iostream>
#include<fstream>
#include<string>
#include <iomanip> // We will use this library for Output Formatting

using namespace std;

class Player_record {
private:
      struct Node
      {
            //Following are the Info about a Player:
            string name;
            int age;
            int score;
            int matches_played;

            //Following are the Pointer of Next and previous:
            Node* next;
            Node* pre;

        };

        void writeHeaderIfNeeded() {
            ifstream file("Record.txt");
            if (!file || file.peek() == ifstream::traits_type::eof()) {
                ofstream outfile("Record.txt");
                outfile << left << setw(8) << "No."
                    << setw(20) << "Name"
                    << setw(10) << "Age"
                    << setw(10) << "Score"
                    << setw(15) << "Matches Played" << endl;
                outfile << string(55, '-') << endl;
                outfile.close();
            }


        }

        typedef Node* NODEPTR;
        NODEPTR head, tail;
        int count;

    public:
        Player_record() {
            head = nullptr;
            tail = nullptr;
            writeHeaderIfNeeded();
            count = 0;
        }

        // ------------------------------------------------------------
        void createNode(string name, int age, int score, int matches_played) {
            NODEPTR NEW_player = new Node();

            NEW_player->name = name;
            NEW_player->age = age;
            NEW_player->score = score;
            NEW_player->matches_played = matches_played;
            NEW_player->next = nullptr;

            if (head == nullptr) {
                NEW_player->pre = nullptr;
                head = tail = NEW_player;
            }
            else {
                NEW_player->pre = tail;
                tail->next = NEW_player;
                tail = NEW_player;
            }
            count++;
        }
        //  -----------------------------------------------------------
        void addPlayerRecord(string name, int age, int score, int matches_played) {
            ofstream file("Record.txt", ios::app);
            if (!file) {
                cerr << "Error while opening for writnig." << endl;
                return;
            }

            createNode(name, age, score, matches_played);

            file << left << setw(8) << count
                << setw(20) << name
                << setw(10)
                << age << setw(10)
                << score << setw(15)
                << matches_played << endl;

            file.close();
            cout << "Player record added successfully!" << endl;


        }


        //  ------------------------------------------------------------
        bool SearchPlayer(string name) {

            NODEPTR current = head;
            while (current != nullptr) {
                if (current->name == name) {
                    return true;
                }
                current = current->next;
            }
            return false;
        }
        //  ------------------------------------------------------------
        void display_players() {

            ifstream file("Record.txt");
            if (!file) {
                cerr << "Error: Unable to open the file!" << endl;
                return;
            }


            string name;
            int age, score, matches_played;

            string header;
            getline(file, header);
            getline(file, header);

            while (file >> name >> age >> score >> matches_played) {
                if (!SearchPlayer(name)) {

                    createNode(name, age, score, matches_played);

                }
            }

            file.close();
```

```
        cout << left << setw(8) << "No. "
            << setw(20) << "Name"
            << setw(10) << "Age"
            << setw(10) << "Score"
            << setw(15) << "Matches Played" << endl;
        cout << string(55, '-') << endl;

        NODEPTR current = head;
        int c = 1;
        while (current != nullptr) {
            cout << left << setw(8) << c
                << setw(20) << current->name
                << setw(10) << current->age
                << setw(10) << current->score
                << setw(15) << current->matches_played << endl;
            current = current->next; c++;
        }
    }
    //  ------------------------------------------------------------

};

void Menu() {
    cout << "---------- Menu ----------" << endl;
    cout << "1.Add Player Record." << endl;

    cout << "2.Display Players Record." << endl;


    cout << "0.Exit" << endl;


}

string InputName() {
    string n;
    cout << "Enter the Player name: ";
    cin.ignore();
    getline(cin, n);
    return n;
}
int Inputage() {
    int age;
    cout << "Enter the Player age: ";
    cin >> age;
    return age;
}
int Inputscore() {
    int score;
    cout << "Enter Player Scores: ";
    cin >> score;
    return score;
}
int InputMatches_played() {
    int Matches;
    cout << "Enter the Played Matches Number By player:  ";
    cin >> Matches;
    return Matches;
```

```
}

int main() {
    Player_record player;
    do
    {
        char option;
        system("cls");
        Menu();
        cin >> option;

        if (option == '1') {
            string name;
            int age, score, matches_played;

            name = InputName();
            age = Inputage();
            score = Inputscore();
            matches_played = InputMatches_played();

            player.addPlayerRecord(name, age, score, matches_played);

        }

        else if (option == '2')
        {
            player.display_players();
            system("pause");
        }

        else if (option == '0')
        {
            exit(1);
        }

        else {
            cout << "Invalid Option !!" << endl;
            system("pause");
        }

    } while (true);



     }
```