**Institute of Computer Technology**

**B. Tech. Computer Science and Engineering**

**Sub: ESFP – II**

**Course Code: 2CSE203**

**Practical – 14**

**Name: Jaymin Gondaliya**

**Enrollment No: 23162171007**

**Sem - 2**

**Branch: CS**

**Class: B**

**Batch: 25**

**Objective:**

To learn and implement object-oriented file handling concept, for create file,open file, write text into file, read text from file, and close file by using some pre-defined classes like (ofstream, ifstream) with some pre-defined mode like (out mode, in mode, app mode, etc) and their function like write (), read (), formatting function and etc.

**Problem Definition-1:** Complete the code for the object assigned to you to satisfy the following specifications.

**Code:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Employee

{

public:

    string name;

    int id;

    double salary;

    string designation;

public:

    void input()

    {

        cout << "Enter name: ";

        cin >> name;

        cout << "Enter ID: ";

        cin >> id;

        cout << "Enter salary: ";

        cin >> salary;

        cout << "Enter designation: ";

        cin >> designation;

    }

    void display()

    {

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

        cout << "ID: " << id << endl;

        cout << "Salary: " << salary << endl;

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

    }

    string getName()

    {

        return name;

    }

    int getID()

    {

        return id;

    }

    double getSalary()

    {

        return salary;

    }

    string getDesignation()

    {

        return designation;

    }

};

void addData()

{

    Employee emp;

    emp.input();

    ofstream file("data.bin", ios::app | ios::binary);

    if (file.is\_open())

    {

        file << emp.getName() << " " << emp.getID() << " " << emp.getSalary() << " " << emp.getDesignation() << endl;

        file.close();

        cout << "Data added successfully!" << endl;

    }

    else

    {

        cout << "Unable to open file." << endl;

    }

}

void viewData()

{

    ifstream file("data.bin",ios::binary);

    if (file.is\_open())

    {

        string name;

        int id;

        double salary;

        string designation;

        while (file >> name >> id >> salary >> designation)

        {

            cout << "Name: " << name << ", ID: " << id << ", Salary: " << salary << ", Designation: " << designation << endl;

        }

        file.close();

    }

    else

    {

        cout << "Unable to open file." << endl;

    }

}

void deleteData()

{

    string name;

    cout << "Enter name to delete: ";

    cin >> name;

    ifstream file("data.bin");

    if (file.is\_open())

    {

        ofstream temp("temp.bin");

        string n;

        int id;

        double salary;

        string designation;

        while (file >> n >> id >> salary >> designation)

        {

            if (n != name)

            {

                temp << n << " " << id << " " << salary << " " << designation << endl;

            }

        }

        file.close();

        temp.close();

        remove("data.bin");

        rename("temp.bin", "data.bin");

        cout << "Data deleted successfully!" << endl;

    }

    else

    {

        cout << "Unable to open file." << endl;

    }

}

void updateData()

{

    string name;

    cout << "Enter name to update: ";

    cin >> name;

    ifstream file("data.bin");

    if (file.is\_open())

    {

        ofstream temp("temp.bin");

        string n;

        int id;

        double salary;

        string designation;

        while (file >> n >> id >> salary >> designation)

        {

            if (n != name)

            {

                temp << n << " " << id << " " << salary << " " << designation << endl;

            }

            else

            {

                Employee emp;

                emp.input();

                temp << emp.getName() << " " << emp.getID() << " " << emp.getSalary() << " " << emp.getDesignation() << endl;

            }

        }

        file.close();

        temp.close();

        remove("data.bin");

        rename("temp.bin", "data.bin");

        cout << "Data updated successfully!" << endl;

    }

    else

    {

        cout << "Unable to open file." << endl;

    }

}

void searchData()

{

    string name;

    cout << "Enter name to search: ";

    cin >> name;

    ifstream file("data.bin");

    if (file.is\_open())

    {

        string n;

        int id;

        double salary;

        string designation;

        bool found = false;

        while (file >> n >> id >> salary >> designation)

        {

            if (n == name)

            {

                cout << "Name: " << n << ", ID: " << id << ", Salary: " << salary << ", Designation: " << designation << endl;

                found = true;

                break;

            }

        }

        file.close();

        if (!found)

        {

            cout << "Data not found." << endl;

        }

    }

    else

    {

        cout << "Unable to open file." << endl;

    }

}

int main()

{

    int choice;

    do

    {

        cout << "1. Add data" << endl;

        cout << "2. View data" << endl;

        cout << "3. Delete data" << endl;

        cout << "4. Update data" << endl;

        cout << "5. Search data" << endl;

        cout << "6. Sort data" << endl;

        cout << "0. Exit" << endl;

        cout << "Enter your choice: ";

        cin >> choice;

        Employee emp[10];

        switch (choice)

        {

        case 1:

            addData();

            break;

        case 2:

            viewData();

            break;

        case 3:

            deleteData();

            break;

        case 4:

            updateData();

            break;

        case 5:

            searchData();

            break;

        case 6:

{

    ifstream file("data.bin");

    if (file.is\_open())

    {

        Employee emp[10];

        int count = 0;

        while (count < 10 && file >> emp[count].name >> emp[count].id >> emp[count].salary >> emp[count].designation)

        {

            count++;

        }

        file.close();

        int input;

        cout << "1. Ascending." << endl;

        cout << "2. Descending." << endl;

        cout << "Enter choice: ";

        cin >> input;

        Employee temp;

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

        {

            for (int j = i + 1; j < count; j++)

            {

                string comp = emp[i].name;

                string comp1 = emp[j].name;

                if ((comp.compare(comp1) > 0 && input == 1) || (comp.compare(comp1) < 0 && input == 2))

                {

                    temp = emp[i];

                    emp[i] = emp[j];

                    emp[j] = temp;

                }

            }

        }

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

        {

            cout << "Name: " << emp[i].name << ", ID: " << emp[i].id << ", Salary: " << emp[i].salary << ", Designation: " << emp[i].designation << endl;

        }

    }

    else

    {

        cout << "Unable to open file." << endl;

    }

    break;

}

        case 0:

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

            break;

        default:

            cout << "Invalid choice. Please try again." << endl;

            break;

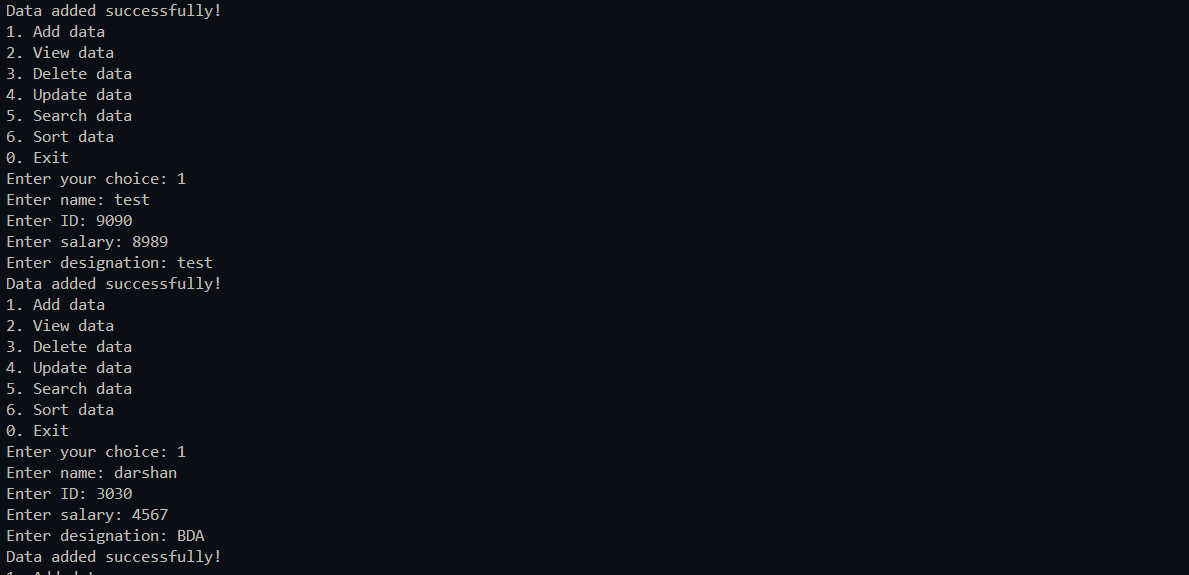
        }

    } while (choice != 0);

    return 0;

}

**Output –**

****

****