Assignment 12

```
#include <iostream>

#include <fstream>
#include <vector>
#include <algorithm>
#include <cstring>
using namespace std;

class Employee {
    int empID;
    char name[50];
    char designation[30];
    float salary;

public:
    void input();
    void display() const;
    int getID() const { return empID; }
    void modify();

    // File operations
    static void addEmployee();
    static void displayEmployee(int);
    static void deleteEmployee(int);
    static void updateEmployee(int);
    static void displayAll();
};

void Employee::input() {
    cout << "Enter Employee ID: ";
    cin >> empID;
    cin.ignore();
    cout << "Enter Name: ";
    cin.getline(name, 50);
    cout << "Enter Designation: ";
    cin.getline(designation, 30);
    cout << "Enter Salary: ";
    cin >> salary;
}

void Employee::display() const {
    cout << "\nID: " << empID
         << "\nName: " << name
         << "\nDesignation: " << designation
         << "\nSalary: " << salary << endl;
}

void Employee::modify() {
    cout << "Modifying Employee ID: " << empID << endl;
    cin.ignore();
```

```
    cout << "Enter new name (or '.' to keep unchanged): ";
    char newName[50];
    cin.getline(newName, 50);
    if (strcmp(newName, ".") != 0)
        strcpy(name, newName);

    cout << "Enter new designation (or '.' to keep unchanged): ";
    char newDesig[30];
    cin.getline(newDesig, 30);
    if (strcmp(newDesig, ".") != 0)
        strcpy(designation, newDesig);

    cout << "Enter new salary (or -1 to keep unchanged): ";
    float newSalary;
    cin >> newSalary;
    if (newSalary != -1)
        salary = newSalary;
}

void Employee::addEmployee() {
    Employee emp;
    emp.input();

    ofstream out("employee.dat", ios::binary | ios::app);
    out.write(reinterpret_cast<char*>(&emp), sizeof(emp));
    out.close();

    cout << "Employee added successfully.\n";
}

void Employee::displayEmployee(int id) {
    ifstream in("employee.dat", ios::binary);
    Employee emp;
    bool found = false;

    while (in.read(reinterpret_cast<char*>(&emp), sizeof(emp))) {
        if (emp.getID() == id) {
            emp.display();
            found = true;
            break;
        }
    }
    in.close();

    if (!found)
        cout << "Employee with ID " << id << " not found.\n";
}

void Employee::deleteEmployee(int id) {
    ifstream in("employee.dat", ios::binary);
    ofstream out("temp.dat", ios::binary);
    Employee emp;
    bool found = false;

    while (in.read(reinterpret_cast<char*>(&emp), sizeof(emp))) {
        if (emp.getID() != id)
            out.write(reinterpret_cast<char*>(&emp), sizeof(emp));
        else
            found = true;
```

```
    }

    in.close();
    out.close();
    remove("employee.dat");
    rename("temp.dat", "employee.dat");

    if (found)
        cout << "Employee deleted successfully.\n";
    else
        cout << "Employee ID not found.\n";
}

void Employee::updateEmployee(int id) {
    fstream file("employee.dat", ios::binary | ios::in | ios::out);
    Employee emp;
    bool found = false;

    while (file.read(reinterpret_cast<char*>(&emp), sizeof(emp))) {
        if (emp.getID() == id) {
            file.seekp(-static_cast<int>(sizeof(emp)), ios::cur);
            emp.modify();
            file.write(reinterpret_cast<char*>(&emp), sizeof(emp));
            found = true;
            break;
        }
    }
    file.close();

    if (found)
        cout << "Employee record updated.\n";
    else
        cout << "Employee ID not found.\n";
}

void Employee::displayAll() {
    ifstream in("employee.dat", ios::binary);
    Employee emp;
    cout << "\nAll Employee Records:\n";
    while (in.read(reinterpret_cast<char*>(&emp), sizeof(emp))) {
        emp.display();
    }
    in.close();
}

int main() {
    int choice, id;

    do {
        cout << "\nEmployee Management Menu\n"
             << "1. Add Employee\n"
             << "2. Display Employee by ID\n"
             << "3. Delete Employee\n"
             << "4. Update Employee\n"
             << "5. Display All Employees\n"
             << "6. Exit\n"
             << "Enter choice: ";
        cin >> choice;
```

```
        switch (choice) {
            case 1:
                Employee::addEmployee();
                break;
            case 2:
                cout << "Enter ID to search: ";
                cin >> id;
                Employee::displayEmployee(id);
                break;
            case 3:
                cout << "Enter ID to delete: ";
                cin >> id;
                Employee::deleteEmployee(id);
                break;
            case 4:
                cout << "Enter ID to update: ";
                cin >> id;
                Employee::updateEmployee(id);
                break;
            case 5:
                Employee::displayAll();
                break;
            case 6:
                cout << "Exiting...\n";
                break;
            default:
                cout << "Invalid choice.\n";
        }
    } while (choice != 6);

    return 0;
}
```

Output:

```
Employee Management Menu
1. Add Employee
2. Display Employee by ID
3. Delete Employee
4. Update Employee
5. Display All Employees
6. Exit
Enter choice: 1
Enter Employee ID: 101
Enter Name: Alice Smith
Enter Designation: Manager
Enter Salary: 65000
Employee added successfully.

Enter choice: 1
Enter Employee ID: 102
Enter Name: Bob Johnson
Enter Designation: Clerk
Enter Salary: 35000
Employee added successfully.

Enter choice: 5

All Employee Records:

ID: 101
Name: Alice Smith
Designation: Manager
Salary: 65000

ID: 102
Name: Bob Johnson
Designation: Clerk
Salary: 35000

Enter choice: 2
Enter ID to search: 101

ID: 101
Name: Alice Smith
Designation: Manager
Salary: 65000

Enter choice: 4
Enter ID to update: 101
Modifying Employee ID: 101
Enter new name (or '.' to keep unchanged): Alicia
Enter new designation (or '.' to keep unchanged): Senior Manager
Enter new salary (or -1 to keep unchanged): 72000
Employee record updated.

Enter choice: 3
Enter ID to delete: 102
Employee deleted successfully.

Enter choice: 5

All Employee Records:

ID: 101
Name: Alicia
Designation: Senior Manager
Salary: 72000

Enter choice: 6
Exiting...
```