

```

#include<iostream>
#include<fstream>
#include<string>
using namespace std;

struct Employee {
    int id;
    string name;
    string designation;
    float salary;

    // Function to serialize Employee object to file
    void writeToFile(ofstream& out) const {
        out.write((char*)&id, sizeof(id));
        size_t nameLen = name.size();
        out.write((char*)&nameLen, sizeof(nameLen));
        out.write(name.c_str(), nameLen);
        size_t desgLen = designation.size();
        out.write((char*)&desgLen, sizeof(desgLen));
        out.write(designation.c_str(), desgLen);
        out.write((char*)&salary, sizeof(salary));
    }

    // Function to deserialize Employee object from file
    bool readFromFile(ifstream& in) {
        if (!in.read((char*)&id, sizeof(id))) return false;
        size_t nameLen;
        in.read((char*)&nameLen, sizeof(nameLen));
        name.resize(nameLen);
        in.read(&name[0], nameLen);
        size_t desgLen;
        in.read((char*)&desgLen, sizeof(desgLen));
        designation.resize(desgLen);
        in.read(&designation[0], desgLen);
        in.read((char*)&salary, sizeof(salary));
        return true;
    }
};

void addEmployee(const string& filename) {
    Employee emp;
    cout << "Enter employee ID: ";
    cin >> emp.id;
    cin.ignore();
}

```

```

    cout << "Enter employee name: ";
    getline(cin, emp.name);
    cout << "Enter employee designation: ";
    getline(cin, emp.designation);
    cout << "Enter employee salary: ";
    cin >> emp.salary;

    ofstream out(filename, ios::binary | ios::app);
    emp.writeToFile(out);
    out.close();
    cout << "Employee added successfully.\n";
}

void deleteEmployee(const string& filename) {
    int delID;
    cout << "Enter ID of employee to delete: ";
    cin >> delID;

    ifstream in(filename, ios::binary);
    ofstream temp("temp.dat", ios::binary);
    bool found = false;
    Employee emp;

    while (emp.readFromFile(in)) {
        if (emp.id != delID) {
            emp.writeToFile(temp);
        } else {
            found = true;
        }
    }
}

in.close();
temp.close();
remove(filename.c_str());
rename("temp.dat", filename.c_str());

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

void displayEmployee(const string& filename) {
    int searchID;

```

```

cout << "Enter ID of employee to display: ";
cin >> searchID;

ifstream in(filename, ios::binary);
Employee emp;
bool found = false;

while (emp.readFromFile(in)) {
    if (emp.id == searchID) {
        found = true;
        cout << "\nEmployee ID: " << emp.id
            << "\nName: " << emp.name
            << "\nDesignation: " << emp.designation
            << "\nSalary: " << emp.salary << "\n";
        break;
    }
}

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

void displayAllEmployees(const string& filename) {
    ifstream in(filename, ios::binary);
    Employee emp;
    bool any = false;

    cout << "\nAll Employees:\n";
    while (emp.readFromFile(in)) {
        any = true;
        cout << "\nEmployee ID: " << emp.id
            << "\nName: " << emp.name
            << "\nDesignation: " << emp.designation
            << "\nSalary: " << emp.salary << "\n";
    }

    in.close();
    if (!any)
        cout << "No employee records found.\n";
}

int main() {
    const string filename = "employee.dat";

```

```
int choice;
do {
    cout << "\n--- Employee Management System ---\n";
    cout << "1. Add Employee\n";
    cout << "2. Delete Employee\n";
    cout << "3. Display Employee by ID\n";
    cout << "4. Display All Employees\n";
    cout << "5. Exit\n";
    cout << "Enter your choice: ";
    cin >> choice;
    cin.ignore(); // flush newline

    switch (choice) {
        case 1: addEmployee(filename); break;
        case 2: deleteEmployee(filename); break;
        case 3: displayEmployee(filename); break;
        case 4: displayAllEmployees(filename); break;
        case 5: cout << "Exiting...\n"; break;
        default: cout << "Invalid choice. Try again.\n";
    }
} while (choice != 5);
return 0;
}
```