

```
#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

class Employee {

private:

int employeeID;

char name[50];

char designation[50];

float salary;

public:

Employee(int id = 0, const char* n = "", const char* desig = "", float sal = 0.0) {

employeeID = id;

strcpy(name, n);

strcpy(designation, desig);

salary = sal;

}

int getEmployeeID() const {

return employeeID;

}

void input() {

cout << "Enter Employee ID: ";

cin >> employeeID;

cin.ignore();

cout << "Enter Name: ";

cin.getline(name, 50);

cout << "Enter Designation: ";

cin.getline(designation, 50);

cout << "Enter Salary: ";

cin >> salary;

}
```

```

void display() const {
    cout << "\nEmployee Details:\n";
    cout << "ID: " << employeeID << endl;
    cout << "Name: " << name << endl;
    cout << "Designation: " << designation << endl;
    cout << "Salary: " << salary << endl;
}

void writeToFile(fstream &file) const {
    file.write((char*)this, sizeof(Employee));
}

void readFromFile(fstream &file) {
    file.read((char*)this, sizeof(Employee));
}

};

void addEmployee(fstream &file) {
    Employee emp;
    emp.input();
    file.seekg(0, ios::beg);

    Employee temp;
    bool exists = false;
    while (file.read((char*)&temp, sizeof(Employee))) {
        if (temp.getEmployeeID() == emp.getEmployeeID()) {
            exists = true;
            break;
        }
    }

    if (exists) {
        cout << "Employee with ID " << emp.getEmployeeID() << " already exists!\n";
    } else {
        file.clear();
        file.seekp(0, ios::end);
    }
}

```

```

emp.writeToFile(file);
cout << "Employee added successfully!\n";
}
}

void deleteEmployee(fstream &file) {
int id;
cout << "Enter Employee ID to delete: ";
cin >> id;

fstream tempFile("temp.dat", ios::out | ios::binary);
file.seekg(0, ios::beg);

Employee emp;
bool found = false;
while (file.read((char*)&emp, sizeof(Employee))) {
if (emp.getEmployeeID() == id) {
found = true;
cout << "Employee with ID " << id << " deleted.\n";
} else {
emp.writeToFile(tempFile);
}
}

file.close();
tempFile.close();

if (!found) {
cout << "Employee with ID " << id << " not found!\n";
} else {
remove("employee_data.dat");
rename("temp.dat", "employee_data.dat");
}

file.open("employee_data.dat", ios::in | ios::out | ios::binary | ios::app);
}

void displayEmployee(fstream &file) {

```

```

int id;

cout << "Enter Employee ID to display: ";

cin >> id;

file.seekg(0, ios::beg);

Employee emp;

bool found = false;

while (file.read((char*)&emp, sizeof(Employee))) {

    if (emp.getEmployeeID() == id) {

        found = true;

        emp.display();

        break;

    }

}

if (!found) {

    cout << "Employee with ID " << id << " not found!\n";

}

}

int main() {

    fstream file("employee_data.dat", ios::in | ios::out | ios::binary | ios::app);

    if (!file) {

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

        return 1;

    }

    int choice;

    do {

        cout << "Menu:\n";

        cout << "1. Add Employee\n";

        cout << "2. Delete Employee\n";

        cout << "3. Display Employee\n";

        cout << "4. Exit\n";

        cout << "Enter your choice: ";

```

```
cin >> choice;
switch (choice) {
case 1:
addEmployee(file);
break;
case 2:
deleteEmployee(file);
break;
case 3:
displayEmployee(file);
break;
case 4:
cout << "Exit\n";
break;
default:
cout << "Invalid choice!\n";
}
} while (choice != 4);
file.close();
return 0;
}
```