**Roll No.COA206**

**Assignment no. 11**

INPUT :  
#include <iostream>

#include <fstream>

#include <map> #include <cstring> using namespace std;

struct Employee {

int id;

char name[50]; char designation[50];

int salary;

};

class EmployeeDatabase { string dataFile = "employees.dat"; map<int, streampos> index;

public:

EmployeeDatabase();

void addEmployee(int id, string name, string designation, int salary); void searchEmployee(int id); void deleteEmployee(int id); void displayAll(); };

EmployeeDatabase::EmployeeDatabase() {

ifstream file(dataFile, ios::binary);

if (!file) {

cout << "Creating new database file.\n"; ofstream newFile(dataFile, ios::binary); newFile.close();

}

file.close();

}

void EmployeeDatabase::addEmployee(int id, string name, string designation, int salary) { if (index.find(id) != index.end()) {

cout << "Employee already exists.\n"; return; }

Employee emp = {id, "", "", salary};

strncpy(emp.name, name.c\_str(), sizeof(emp.name) - 1); strncpy(emp.designation, designation.c\_str(), sizeof(emp.designation) - 1);

ofstream file(dataFile, ios::binary | ios::app); streampos pos = file.tellp();

file.write(reinterpret\_cast<char\*>(&emp), sizeof(emp)); file.close(); index[id] = pos;

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

}

void EmployeeDatabase::searchEmployee(int id) {

if (index.find(id) == index.end()) {

cout << "Employee not found.\n"; return;

}

ifstream file(dataFile, ios::binary);

file.seekg(index[id]); Employee emp;

file.read(reinterpret\_cast<char\*>(&emp), sizeof(emp)); file.close();

cout << "ID: " << emp.id << "\nName: " << emp.name << "\nDesignation: " << emp.designation << "\nSalary: Rs. " << emp.salary << endl;

}

void EmployeeDatabase::deleteEmployee(int id) { if (index.erase(id)) {

cout << "Employee deleted.\n";

} else {

cout << "Employee not found.\n";

}

}

void EmployeeDatabase::displayAll() {

ifstream file(dataFile, ios::binary); Employee emp;

while (file.read(reinterpret\_cast<char\*>(&emp), sizeof(emp))) {

if (index.find(emp.id) != index.end()) {

cout << emp.id << " | " << emp.name << " | " << emp.designation << " | Rs. " << emp.salary << endl;

} }

file.close();

}

int main() {

EmployeeDatabase db; int choice, id, salary; string name, designation;

do {

cout << "\n1 -> Add Employee\n2 -> Search Employee\n3 -> Delete Employee\n4 -> Display

All\n5 -> Exit\nChoice: "; cin >> choice; cin.ignore();

switch (choice) { case 1:

cout << "ID: "; cin >> id; cin.ignore(); cout << "Name: "; getline(cin, name); cout << "Designation: "; getline(cin, designation); cout << "Salary: "; cin >> salary; db.addEmployee(id, name, designation, salary); break; case 2:

cout << "ID: "; cin >> id;

db.searchEmployee(id); break; case 3:

cout << "ID: "; cin >> id; db.deleteEmployee(id);

break; case 4: db.displayAll(); break; case 5: cout << "Exiting...\n";

break; default:

cout << "Invalid choice.\n";

}

} while (choice != 5);

return 0;

}

OUTPUT :



