

### Third Year B. Tech (EL & CE)

**Semester: V**

**Subject: Object-Oriented Programming Lab**

**Name: Shreerang Mhatre**

**Class: SY**

**Roll No: 52**

**Batch: A3**

### Experiment No: 02

**Name of the Experiment: Database of Airline Company**

**Performed on: 13/09/2023**

**Submitted on: 27/08/2023**

#### Problem Statement:

Develop an object oriented program in C++ to create a database of employee into system containing following info:

Employee name, Employee No, Qualification ,address ,contact , Salary (basic, DA, Ta , Net Salary)  
Construct the Database with suitable inline member function for initializing and destroying the data via Constructor, default Constructor, Copy Constructor , destructor.

Use dynamic memory allocation concept while creating and destroying the object of class.

Use static data member concept whenever needed display the Employee info.

#### OUTPUT:

```
Employee Name: Shreerang Mhatre
Employee Number: 302
Qualification: B.Tech Engineering
Address: MIT-WPU
Contact Number: +91 123456789
Basic Salary: 5000
DA: 1000
TA: 500
Net Salary: 6500
-----
Employee Name: Aman Singh
Employee Number: 401
Qualification: M.Tech Engineering
Address: BVP PUNE
Contact Number: +91 325689741
Basic Salary: 6000
DA: 1200
TA: 600
Net Salary: 7800
-----
Total Employees: 2
```

Code:

```
#include <iostream>
```

```
#include <string>
```

```
class Employee {
```

```
private:
```

```
    static int empCounter;
```

```
    std::string empName;
```

```
    int empNumber;
```

```
    std::string qualification;
```

```
    std::string address;
```

```
    std::string contactNumber;
```

```
    struct Salary {
```

```
        double basic;
```

```
        double DA;
```

```
        double TA;
```

```
        double netSalary;
```

```
    } salary;
```

```
public:
```

```
    // Parameterized constructor
```

```
    Employee(const std::string& name, int number, const std::string& qual, const std::string& addr,  
            const std::string& contact, double basicSalary, double DA, double TA) {
```

```
        empName = name;
```

```
        empNumber = number;
```

```
        qualification = qual;
```

```
        address = addr;
```

```
        contactNumber = contact;
```

```
        salary.basic = basicSalary;
```

```
        salary.DA = DA;
```

```
        salary.TA = TA;
```

```
        salary.netSalary = calculateNetSalary();
```

```
        empCounter++;
```

```
    }
```

```
    // Default constructor
```

```
    Employee() : Employee("", 0, "", "", "", 0.0, 0.0, 0.0) {}
```

// Copy constructor

```
Employee(const Employee& other) {  
    empName = other.empName;  
    empNumber = other.empNumber;  
    qualification = other.qualification;  
    address = other.address;  
    contactNumber = other.contactNumber;  
    salary.basic = other.salary.basic;  
    salary.DA = other.salary.DA;  
    salary.TA = other.salary.TA;  
    salary.netSalary = other.salary.netSalary;  
    empCounter++;  
}
```

// Destructor

```
~Employee() {  
    empCounter--;  
}
```

// Calculate the net salary based on basic, DA, and TA

```
double calculateNetSalary() const {  
    return salary.basic + salary.DA + salary.TA;  
}
```

// Display employee information

```
void displayInfo() const {  
    std::cout << "Employee Name: " << empName << std::endl;  
    std::cout << "Employee Number: " << empNumber << std::endl;  
    std::cout << "Qualification: " << qualification << std::endl;  
    std::cout << "Address: " << address << std::endl;  
    std::cout << "Contact Number: " << contactNumber << std::endl;  
    std::cout << "Basic Salary: " << salary.basic << std::endl;  
    std::cout << "DA: " << salary.DA << std::endl;  
    std::cout << "TA: " << salary.TA << std::endl;  
    std::cout << "Net Salary: " << salary.netSalary << std::endl;  
    std::cout << "-----" << std::endl;  
}
```

// Static member function to get the total number of employees

```
static int getTotalEmployees() {  
    return empCounter;  
}
```

```
}  
};
```

```
// Initialize the static member
```

```
int Employee::empCounter = 0;
```

```
int main() {
```

```
    // Create employee objects using dynamic memory allocation
```

```
    Employee* emp1 = new Employee("Shreerang Mhatre", 302, "B.Tech Engineering", "MIT-WPU", "+91  
123456789 ", 5000.0, 1000.0, 500.0);
```

```
    Employee* emp2 = new Employee("Aman Singh", 401, "M.Tech Engineering", "BVP PUNE", "+91  
325689741", 6000.0, 1200.0, 600.0);
```

```
    // Display employee information
```

```
    emp1->displayInfo();
```

```
    emp2->displayInfo();
```

```
    // Get the total number of employees
```

```
    std::cout << "Total Employees: " << Employee::getTotalEmployees() << std::endl;
```

```
    // Clean up memory and release resources
```

```
    delete emp1;
```

```
    delete emp2;
```

```
    return 0;
```

```
}
```

```
File Edit Selection View Go Run Terminal Help exp7
exp7.c++ exp_2.c++ X
D:\Object Oriented Programming\exp2> exp_2.c++ > main()
1 /*
2 Employee Data Base
3
4 Develop an object oriented program in C++ to create a database of employee into system containing following info:
5 Employee name, Employee No, Qualification ,address ,contact , Salary (basic, DA, TA , Net Salary)
6 Construct the Database with suitable inline member function for initializing and destroying the data via
7 constructor, default constructor, copy constructor , destructor.
8 Use dynamic memory allocation concept while creating and destroying the object of class.
9 Use static data member concept whenever needed display the Employee info.
10
11 */
12
13 #include <iostream>
14 #include <string>
15
16 class Employee {
17 private:
18     static int empCounter;
19     std::string empName;
20     int empNumber;
21     std::string qualification;
22     std::string address;
23     std::string contactNumber;
24     struct Salary {
25         double basic;
26         double DA;
27         double TA;
28         double netSalary;
29     } salary;
30
31 public:
32     // Parameterized constructor
33     Employee(const std::string& name, int number, const std::string& qual, const std::string& addr,
34             const std::string& contact, double basicSalary, double DA, double TA) {
35         empName = name;
36         empNumber = number;
37         qualification = qual;
38
39         Employee Name: Shreerang Mhatre
40         Employee Number: 302
41         Qualification: B.Tech Engineering
42         Address: MIT-WPU
43         Contact Number: +91 123456789
44         Basic Salary: 5000
45         DA: 1000
46         TA: 500
47         Net Salary: 6500
48
49         Employee Name: Aman Singh
50         Employee Number: 401
51         Qualification: M.Tech Engineering
52         Address: BVP PUNE
53         Contact Number: +91 325689741
54         Basic Salary: 6000
55         DA: 1200
56         TA: 600
57         Net Salary: 7800
58
59         Total Employees: 2
60
61 PS D:\Object Oriented Programming\exp2> cd "d:\Object O
62 riented Programming\exp2\" ; if ($?) { g++ exp_2.c++ -o
63 exp_2 } ; if ($?) { .\exp_2 }
```

```
File Edit Selection View Go Run Terminal Help exp7
exp7.c++ exp_2.c++ X
D:\Object Oriented Programming\exp2> exp_2.c++ > main()
28 double netSalary;
29 } salary;
30
31 public:
32 // Parameterized constructor
33 Employee(const std::string& name, int number, const std::string& qual, const std::string& addr,
34         const std::string& contact, double basicSalary, double DA, double TA) {
35     empName = name;
36     empNumber = number;
37     qualification = qual;
38     address = addr;
39     contactNumber = contact;
40     salary.basic = basicSalary;
41     salary.DA = DA;
42     salary.TA = TA;
43     salary.netSalary = calculateNetSalary();
44     empCounter++;
45 }
46
47 // Default constructor
48 Employee() : Employee("", 0, "", "", 0.0, 0.0, 0.0) {}
49
50 // Copy constructor
51 Employee(const Employee& other) {
52     empName = other.empName;
53     empNumber = other.empNumber;
54     qualification = other.qualification;
55     address = other.address;
56     contactNumber = other.contactNumber;
57     salary.basic = other.salary.basic;
58     salary.DA = other.salary.DA;
59     salary.TA = other.salary.TA;
60     salary.netSalary = other.salary.netSalary;
61     empCounter++;
62 }
63
64 // Destructor
```

```

File Edit Selection View Go Run Terminal Help
exp7

exp7.c++ exp2.c++ X
D:\> Object Oriented Programming > exp2 > exp2.c++ > main()

61     empCounter++;
62 }
63
64 // Destructor
65 ~Employee() {
66     empCounter--;
67 }
68
69 // Calculate the net salary based on basic, DA, and TA
70 double calculateNetSalary() const {
71     return salary.basic + salary.DA + salary.TA;
72 }
73
74 // Display employee information
75 void displayInfo() const {
76     std::cout << "Employee Name: " << empName << std::endl;
77     std::cout << "Employee Number: " << empNumber << std::endl;
78     std::cout << "Qualification: " << qualification << std::endl;
79     std::cout << "Address: " << address << std::endl;
80     std::cout << "Contact Number: " << contactNumber << std::endl;
81     std::cout << "Basic Salary: " << salary.basic << std::endl;
82     std::cout << "DA: " << salary.DA << std::endl;
83     std::cout << "TA: " << salary.TA << std::endl;
84     std::cout << "Net Salary: " << salary.netSalary << std::endl;
85     std::cout << "-----" << std::endl;
86 }
87
88 // Static member function to get the total number of employees
89 static int getTotalEmployees() {
90     return empCounter;
91 }
92 };
93
94 // Initialize the static member
95 int Employee::empCounter = 0;
96
97 int main() {
98
99     // Create employee objects using dynamic memory allocation
100     Employee* emp1 = new Employee("Shreerang Mhatre", 302, "B.Tech Engineering", "MIT-WPU", "+91 123456789 ", 5000);
101     Employee* emp2 = new Employee("Aman Singh", 401, "M.Tech Engineering", "BVP PUNE", "+91 325689741", 6000.0, 1200);
102
103     // Display employee information
104     emp1->displayInfo();
105     emp2->displayInfo();
106
107     // Get the total number of employees
108     std::cout << "Total Employees: " << Employee::getTotalEmployees() << std::endl;
109
110     // Clean up memory and release resources
111     delete emp1;
112     delete emp2;
113
114     return 0;
115 }

```

```

PS D:\Object Oriented Programming\exp7> cd "d:\Object O
riented Programming\exp2\"; if ($?) { g++ exp2.c++ -o
exp2 }; if ($?) { .\exp2 }
Employee Name: Shreerang Mhatre
Employee Number: 302
Qualification: B.Tech Engineering
Address: MIT-WPU
Contact Number: +91 123456789
Basic Salary: 5000
DA: 1000
TA: 500
Net Salary: 6500
-----
Employee Name: Aman Singh
Employee Number: 401
Qualification: M.Tech Engineering
Address: BVP PUNE
Contact Number: +91 325689741
Basic Salary: 6000
DA: 1200
TA: 600
Net Salary: 7800
-----
Total Employees: 2
PS D:\Object Oriented Programming\exp2>

```

Ln 114, Col 2 Spaces: 4 UTF-8 CRLF {} C++ Win32 09:39 AM 27-09-2023

```

File Edit Selection View Go Run Terminal Help
exp7

exp7.c++ exp2.c++ X
D:\> Object Oriented Programming > exp2 > exp2.c++ > main()

81     std::cout << "Basic Salary: " << salary.basic << std::endl;
82     std::cout << "DA: " << salary.DA << std::endl;
83     std::cout << "TA: " << salary.TA << std::endl;
84     std::cout << "Net Salary: " << salary.netSalary << std::endl;
85     std::cout << "-----" << std::endl;
86 }
87
88 // Static member function to get the total number of employees
89 static int getTotalEmployees() {
90     return empCounter;
91 }
92 };
93
94 // Initialize the static member
95 int Employee::empCounter = 0;
96
97 int main() {
98     // Create employee objects using dynamic memory allocation
99     Employee* emp1 = new Employee("Shreerang Mhatre", 302, "B.Tech Engineering", "MIT-WPU", "+91 123456789 ", 5000);
100     Employee* emp2 = new Employee("Aman Singh", 401, "M.Tech Engineering", "BVP PUNE", "+91 325689741", 6000.0, 1200);
101
102     // Display employee information
103     emp1->displayInfo();
104     emp2->displayInfo();
105
106     // Get the total number of employees
107     std::cout << "Total Employees: " << Employee::getTotalEmployees() << std::endl;
108
109     // Clean up memory and release resources
110     delete emp1;
111     delete emp2;
112
113     return 0;
114 }
115

```

```

PS D:\Object Oriented Programming\exp7> cd "d:\Object O
riented Programming\exp2\"; if ($?) { g++ exp2.c++ -o
exp2 }; if ($?) { .\exp2 }
Employee Name: Shreerang Mhatre
Employee Number: 302
Qualification: B.Tech Engineering
Address: MIT-WPU
Contact Number: +91 123456789
Basic Salary: 5000
DA: 1000
TA: 500
Net Salary: 6500
-----
Employee Name: Aman Singh
Employee Number: 401
Qualification: M.Tech Engineering
Address: BVP PUNE
Contact Number: +91 325689741
Basic Salary: 6000
DA: 1200
TA: 600
Net Salary: 7800
-----
Total Employees: 2
PS D:\Object Oriented Programming\exp2>

```

Ln 114, Col 2 Spaces: 4 UTF-8 CRLF {} C++ Win32 09:39 AM 27-09-2023