**Name:-** Abitha S

**Class:-**SY-2

**PRN:-**B25CE2015

**Assignment No:-2**

**Title:** Employee Information System

**Problem Statement**:-Develop a program in C++ to create a database of an employee's information system containing the following fields: Name, employee ID,Department, Date of Joining, Contact address, Telephone number etc. Construct the database with suitable member functions to accept and print employee details. Make use of constructor types, destructor, static members, inline function and dynamic memory allocation using operators-new and delete.

**INPUT:-**

#include <iostream>

#include <string>

using namespace std;

class Employee {

private:

string name;

int emp\_id;

string department;

string date\_of\_joining;

string address;

string telephone;

static int employeeCount;

public:

// Default Constructor

Employee() {

name = "Unknown";

emp\_id = 0;

department = "Not Assigned";

date\_of\_joining = "Not Set";

address = "Not Provided";

telephone = "000-0000";

employeeCount++;

cout << "Default constructor called for employee ID: " << emp\_id << endl;

}

// Parameterized Constructor

Employee(string n, int id, string dept, string doj, string addr, string tel) {

name = n;

emp\_id = id;

department = dept;

date\_of\_joining = doj;

address = addr;

telephone = tel;

employeeCount++;

cout << "Parameterized constructor called for: " << name << endl;

}

// Copy Constructor

Employee(const Employee &e) {

name = e.name + " (Copy)";

emp\_id = e.emp\_id + 1000;

department = e.department;

date\_of\_joining = e.date\_of\_joining;

address = e.address;

telephone = e.telephone;

employeeCount++;

cout << "Copy constructor called for: " << name << endl;

}

// Destructor

~Employee() {

cout << "Destructor called for employee: " << name << " (ID: " << emp\_id << ")" << endl;

employeeCount--;

}

// Member function to accept employee details

void acceptDetails() {

cout << "\n--- Enter Employee Details ---" << endl;

cout << "Name: ";

cin.ignore();

getline(cin, name);

cout << "Employee ID: ";

cin >> emp\_id;

cout << "Department: ";

cin >> department;

cout << "Date of Joining (DD-MM-YYYY): ";

cin >> date\_of\_joining;

cout << "Address: ";

cin.ignore();

getline(cin, address);

cout << "Telephone Number: ";

getline(cin, telephone);

}

// Member function to display employee details

void displayDetails() {

cout << "\n--- Employee Details ---" << endl;

cout << "Name: " << name << endl;

cout << "Employee ID: " << emp\_id << endl;

cout << "Department: " << department << endl;

cout << "Date of Joining: " << date\_of\_joining << endl;

cout << "Address: " << address << endl;

cout << "Telephone: " << telephone << endl;

cout << "------------------------" << endl;

}

// Inline function to get employee name

inline string getName() {

return name;

}

// Inline function to get employee ID

inline int getEmpId() {

return emp\_id;

}

// Inline function to get department

inline string getDepartment() {

return department;

}

// Static function to show total employees

static void showTotalEmployees() {

cout << "\nTotal Employees in Database: " << employeeCount << endl;

}

};

// Initialize static member

int Employee::employeeCount = 0;

int main() {

cout << "=== EMPLOYEE INFORMATION SYSTEM ===" << endl;

// Employee 1 - Default constructor

cout << "\n--- Creating Employee 1 (Default Constructor) ---" << endl;

Employee emp1;

emp1.displayDetails();

// Employee 2 - User input for parameterized constructor

cout << "\n--- Creating Employee 2 (Parameterized Constructor) ---" << endl;

string name, dept, doj, addr, tel;

int id;

cout << "Enter details for Employee 2:" << endl;

cout << "Name: ";

getline(cin, name);

cout << "Employee ID: ";

cin >> id;

cout << "Department: ";

cin >> dept;

cout << "Date of Joining (DD-MM-YYYY): ";

cin >> doj;

cout << "Address: ";

cin.ignore();

getline(cin, addr);

cout << "Telephone: ";

getline(cin, tel);

Employee emp2(name, id, dept, doj, addr, tel);

emp2.displayDetails();

// Employee 3 - Copy constructor

cout << "\n--- Creating Employee 3 (Copy Constructor from Employee 2) ---" << endl;

Employee emp3 = emp2;

emp3.displayDetails();

// Employee 4 - Dynamic allocation with user input

cout << "\n--- Creating Employee 4 (Dynamic Allocation) ---" << endl;

Employee \*emp4 = new Employee();

cout << "Enter details for Employee 4:" << endl;

emp4->acceptDetails();

emp4->displayDetails();

// Employee 5 - Another dynamic allocation

cout << "\n--- Creating Employee 5 (Dynamic Allocation) ---" << endl;

Employee \*emp5 = new Employee();

cout << "Enter details for Employee 5:" << endl;

emp5->acceptDetails();

emp5->displayDetails();

// Using inline functions

cout << "\n--- Using Inline Functions ---" << endl;

cout << "Employee 2 Name: " << emp2.getName() << endl;

cout << "Employee 2 ID: " << emp2.getEmpId() << endl;

cout << "Employee 2 Department: " << emp2.getDepartment() << endl;

// Show total employees

Employee::showTotalEmployees();

// Delete dynamic objects

cout << "\n--- Deleting Dynamic Employees ---" << endl;

delete emp4;

delete emp5;

// Final count

cout << "\n--- Final Employee Count ---" << endl;

Employee::showTotalEmployees();

cout << "\n=== Program Ending ===" << endl;

return 0;

}

**OUTPUT:-**

=== EMPLOYEE INFORMATION SYSTEM ===

--- Creating Employee 1 (Default Constructor) ---

Default constructor called for employee ID: 0

--- Employee Details ---

Name: Unknown

Employee ID: 0

Department: Not Assigned

Date of Joining: Not Set

Address: Not Provided

Telephone: 000-0000

------------------------

--- Creating Employee 2 (Parameterized Constructor) ---

Enter details for Employee 2:

Name: Raj Sharma

Employee ID: 101

Department: IT

Date of Joining (DD-MM-YYYY): 15-01-2023

Address: 123 MG Road, Pune

Telephone: 9876543210

Parameterized constructor called for: Raj Sharma

--- Employee Details ---

Name: Raj Sharma

Employee ID: 101

Department: IT

Date of Joining: 15-01-2023

Address: 123 MG Road, Pune

Telephone: 9876543210

------------------------

--- Creating Employee 3 (Copy Constructor from Employee 2) ---

Copy constructor called for: Raj Sharma (Copy)

--- Employee Details ---

Name: Raj Sharma (Copy)

Employee ID: 1101

Department: IT

Date of Joining: 15-01-2023

Address: 123 MG Road, Pune

Telephone: 9876543210

------------------------

--- Creating Employee 4 (Dynamic Allocation) ---

Default constructor called for employee ID: 0

Enter details for Employee 4:

--- Enter Employee Details ---

Name: Priya Patel

Employee ID: 102

Department: HR

Date of Joining (DD-MM-YYYY): 20-02-2023

Address: 456 FC Road, Mumbai

Telephone: 9123456789

--- Employee Details ---

Name: Priya Patel

Employee ID: 102

Department: HR

Date of Joining: 20-02-2023

Address: 456 FC Road, Mumbai

Telephone: 9123456789

------------------------

--- Creating Employee 5 (Dynamic Allocation) ---

Default constructor called for employee ID: 0

Enter details for Employee 5:

--- Enter Employee Details ---

Name: Amit Kumar

Employee ID: 103

Department: Finance

Date of Joining (DD-MM-YYYY): 10-03-2023

Address: 789 JM Road, Delhi

Telephone: 9988776655

--- Employee Details ---

Name: Amit Kumar

Employee ID: 103

Department: Finance

Date of Joining: 10-03-2023

Address: 789 JM Road, Delhi

Telephone: 9988776655

------------------------

--- Using Inline Functions ---

Employee 2 Name: Raj Sharma

Employee 2 ID: 101

Employee 2 Department: IT

Total Employees in Database: 5

--- Deleting Dynamic Employees ---

Destructor called for employee: Priya Patel (ID: 102)

Destructor called for employee: Amit Kumar (ID: 103)

--- Final Employee Count ---

Total Employees in Database: 3

=== Program Ending ===

Destructor called for employee: Raj Sharma (Copy) (ID: 1101)

Destructor called for employee: Raj Sharma (ID: 101)

Destructor called for employee: Unknown (ID: 0)