

CS224 – Fall 2025 - Lab Quiz 03

Question 1

Problem Statement

You are required to implement a simple **Employee Management System** for Habib.

The system should consist of a base class **Employee** and two derived classes **Staff** and **Faculty**. Each time a new employee (either Staff or Faculty) is created, a shared static counter should keep track of the total number of employees created so far.

Input Format

The program should take input from the user as follows:

- The first line contains the number of employees, n .
- The subsequent n lines each contain the type of employee, the employee name, and their salary.

Output Format

The program should display information about each employee and print the total number of employees .

Function Description(s)

You must implement the following classes and their member functions:

- **class Employee**
 - `protected: string name;` — stores the employee's name.
 - `protected: double salary;` — stores the employee's salary.
 - `static int employeeCount;` — shared among all employees to count total employees.
 - `Employee(string n, double s);` — constructor that initializes name and salary, and increments `employeeCount`.
 - `virtual void displayInfo();` — prints the employee's name and salary.
 - `static void showTotalEmployees();` — displays total number of employees created.
 - `virtual ~Employee();` — virtual destructor for safe deletion of derived objects.
- **class Staff : public Employee**
 - `private: string department;` - stores the department that the Staff is a part of.
 - Overrides `displayInfo()` to include the title "(Staff)".
- **class Faculty : public Employee**
 - `private: string department;` - stores the department that the Faculty is a part of.
 - Overrides `displayInfo()` to include the title "(Faculty)".

- `int main()`
 - Reads input for the number of employees, n .
 - Reads the next n lines for information on each employee (staff or faculty, name, and salary).
 - Creates the n objects dynamically.
 - Stores their pointers in an array of `Employee*`.
 - Uses a loop to call `displayInfo()` on each element in the array of `Employee*`.
 - Calls `Employee::showTotalEmployees()` to display the total count.
 - Deletes dynamically allocated objects to prevent memory leaks.

Constraints

- Use inheritance to derive `Staff` and `Faculty` from `Employee`.
- Use virtual functions for runtime polymorphism.
- Use a static variable to track total employees.
- All output formatting must match the sample output exactly.
- Use dynamic memory allocation (`new` and `delete`) for objects.

Example

Input

```
>>> 2
>>> Staff Ali 75000
>>> Faculty Salman 80000
```

Output

```
Name: Ali (Staff), Salary: 75000
Name: Salman (Faculty), Salary: 80000
Total Employees Created: 2
```