**OBJECT-ORIENTED PROGRAMMING**

|  |  |
| --- | --- |
| Lab 7 | |
| **Topic** | Operator overloading, static and friend functions |
| **Objective** | To learn the usage of pointers and arrays and Functions |

**Task 1: () Employee Management System**

Create a simple Employee Management System in C++. The system should have the following features:

Employee Class:

Create an **“Employee”** class with private attributes for the **employee's name, employee ID, and a static** attribute **Noe** to keep track of the total number of employees.

make **a constructor** that takes the employee's name and initializes the ID. The ID should be automatically assigned and incremented for each new employee.

Implement a **static function** to get the total number of employees.

**Department Class:**

Create a “**Department”** class with private attributes for the department name and a list of employees.

Write a function to add an employee to the department.

Implement a **friend function** that prints the details of all employees in a given department.

**Main Function:**

In the main function, create a few instances of the Employee class and add them to different departments.

Print the total number of employees using the **static function** in the Employee class.

Use the **friend function** to print the details of employees in a specific department.

**Task 2: Vector Addition**

Create a **Vector** class to represent 3D vectors. Overload the **+** operator to add two vectors. Create a program that demonstrates vector addition.

**Task 3: Matrix Multiplication**

Create a **Matrix** class to represent **nxn** matrices. Overload the **\*** operator to multiply two matrices. Create a program that demonstrates matrix multiplication.

**Task 4: Complex Number Operations**

Create a **Complex** class to represent complex numbers. Overload the basic arithmetic operators (**+**, **-**, **\***, **/**) to perform operations on complex numbers. Create a program that demonstrates various complex number operations.

**Task 6: Date Comparison**

Create a **Date** class to represent a date (day, month, year). Overload the **==** and **!=** operators to compare two **Date** objects for equality and inequality. Create a program that demonstrates date comparison.