

Note:

1. This assignment is designed to practice static fields, static initializers, and static methods.
 2. Understand the problem statement and use static and non-static wisely to solve the problem.
 3. Use constructors, proper getter/setter methods, and `toString()` wherever required.
1. Design and implement a class named `InstanceCounter` to track and count the number of instances created from this class.
 2. Design and implement a class named `Logger` to manage logging messages for an application. The class should be implemented as a singleton to ensure that only one instance of the `Logger` exists throughout the application.

The class should include the following methods:

- `getInstance()`: Returns the unique instance of the `Logger` class.
 - `log(String message)`: Adds a log message to the logger.
 - `getLog()`: Returns the current log messages as a `String`.
 - `clearLog()`: Clears all log messages.
3. Design and implement a class named `Employee` to manage employee data for a company. The class should include fields to keep track of the total number of employees and the total salary expense, as well as individual employee details such as their ID, name, and salary.

The class should have methods to:

- Retrieve the total number of employees (`getTotalEmployees()`)
- Apply a percentage raise to the salary of all employees (`applyRaise(double percentage)`)
- Calculate the total salary expense, including any raises (`calculateTotalSalaryExpense()`)
- Update the salary of an individual employee (`updateSalary(double newSalary)`)

Understand the problem statement and use static and non-static fields and methods appropriately. Implement static and non-static initializers, constructors, getter and setter methods, and a `toString()` method to handle the initialization and representation of employee data.

Write a menu-driven program in the `main` method to test the functionalities.