

Error Log and Fixes

Error 1

- Issue:
The `Faculty`, `Staff`, and `UEmployee` classes were nested inside the `Employee` class, making the code unnecessarily complex and harder to maintain.
- Fix:
I refactored the code by moving the `Faculty`, `Staff`, and `UEmployee` classes into separate standalone files. This improved modularity, allowing each class to be maintained independently and reused in other contexts without modification.

Error 2

- Issue:
The `UEmployee` constructor did not validate salary input, allowing negative salary values to be assigned to employees.
- Fix:
I added a validation check to ensure the salary is non-negative. If a negative salary is provided, the constructor throws an `IllegalArgumentException`, ensuring data validity and preventing errors.

Error 3

- Issue:
Polymorphism was not utilized effectively. I treated `Faculty` and `Staff` as separate entities, resulting in redundant code.
- Fix:
I applied polymorphism by treating `Faculty` and `Staff` as `UEmployee` objects where possible. This reduced redundancy by allowing the use of a `UEmployee` reference to handle both `Faculty` and `Staff` objects, enhancing code flexibility.

Error 4

- Issue:
Employee information was manually constructed in multiple places, making the code repetitive and difficult to maintain.
- Fix:
I introduced a `toString` method in the `UEmployee`, `Faculty`, and `Staff` classes to

centralize and standardize the formatting of employee details. This streamlined the code and made future updates easier.