**Object Oriented Programming**

**Class Assignment # 4**

**Total Marks: 20** **Deadline: 26-01-25: till 11:59 PM**

**Instructor Name: Hira Naveed**

|  |  |  |  |
| --- | --- | --- | --- |
| **CLO No.** | **Course Learning Outcome (CLO)** | **Taxonomy Level** | **Mapping to PLO** |
| **CLO 2** | Identify the objects & their relationships to build object-oriented solution. | C3 | 3 |

**Hotel Management System**

**Scenario:**

You are tasked with developing a **Hotel Management System** for a chain of hotels. The system will manage different types of employees, their roles, and salaries, as well as calculate the revenue generated by the hotel. The system should be designed using **inheritance**, **polymorphism**, **interfaces**, and **multi-level inheritance**. Additionally, a simple **GUI** should allow the user to interact with the system by adding employees, calculating their salaries, and displaying hotel revenue.Also create its **UML diagram**

**Requirements:**

**Core Classes and Relationships:**

1. **Employee (Abstract Base Class):**
   * Attributes: name, age, baseSalary.
   * Methods:
     + calculateSalary() (abstract method): Calculates the total salary for the employee.
     + getDetails(): Returns the details of the employee.
     + Getter and setter methods.
2. **Manager (Derived Class from Employee):**
   * Additional Attribute: bonus.
   * Overrides calculateSalary() to include the bonus.
3. **HousekeepingStaff (Derived Class from Employee):**
   * Additional Attribute: hoursWorked.
   * Overrides calculateSalary() to include payment based on hours worked.
4. **Chef (Derived Class from Employee):**
   * Additional Attribute: specialization.
   * Overrides calculateSalary() to include a performance bonus based on the specialization.
5. **Revenue (Interface):**
   * Method: generateRevenue(): Calculates and returns the revenue generated by the hotel.
6. **Hotel (Implements Revenue):**
   * Attributes: hotelName, location, employeeList (list of Employee objects).
   * Implements generateRevenue() based on a fixed value per employee.
   * Methods to add employees and display their details.

**GUI Requirements:**

The GUI should include the following components:

1. **Add Employee:**
   * Input fields to take:
     + Name (JTextField)
     + Age (JTextField)
     + Role (JComboBox: Manager, Housekeeping Staff, Chef)
     + Salary-related details (JTextField)
   * A JButton to add the employee to the hotel's employee list.
2. **Calculate Salary:**
   * Input field for employee name (JTextField).
   * A JButton to calculate and display the salary of the selected employee in a JTextArea.
3. **View Hotel Revenue:**
   * A JButton to calculate and display the total revenue of the hotel in a JTextArea.
4. **Display All Employees:**
   * A JButton to display all employees and their details in a JTextArea.

**Functional Requirements:**

1. **Polymorphism:**
   * Use calculateSalary() to demonstrate polymorphism.
   * Different salary calculations for each employee type.
2. **Inheritance:**
   * Base class Employee inherited by Manager, HousekeepingStaff, and Chef.
3. **Interface:**
   * Implement Revenue in the Hotel class to calculate revenue.
4. **Multi-level Inheritance:**
   * Add an additional class, SeniorManager (derived from Manager), with extra functionality for decision-making bonus.
5. **Validation:**
   * Prevent duplicate employee names.
   * Ensure age and salary inputs are numeric.