**Description of the Employee Management System**

This Python code implements a simplified employee management system that models different types of employees in an organization. The system is built using object-oriented programming principles and includes classes for various employee roles such as full-time employees, part-time employees, contractors, and interns. Each class inherits from a base Employee class, which provides common attributes and methods shared among all employee types.

**Key Components of the System:**

1.**Employee Base Class**:

• **Purpose**: The Employee class serves as the base class for all types of employees. It includes basic attributes like name, employee\_id, performance\_rating, status, and benefits. It also contains methods that handle common operations such as setting performance ratings, adjusting salaries, updating employee status, adding benefits, and displaying employee information.

• **Methods**:

• calculate\_salary(): A placeholder method intended to be overridden in derived classes to calculate the salary specific to the employee type.

• set\_performance\_rating(rating): Sets the performance rating of the employee.

• adjust\_salary(adjustment): Adjusts the employee’s salary by a specified amount.

• update\_status(new\_status): Updates and prints the employee’s current status (e.g., Active, On Leave, Terminated).

• add\_benefit(benefit\_name): Adds a benefit to the employee’s benefits package and prints the addition status.

• display\_info(): Displays and prints the basic information of the employee, including name, ID, salary, performance rating, status, and benefits.

2.**FullTimeEmployee Class**:

• **Purpose**: The FullTimeEmployee class represents full-time employees who are salaried and entitled to vacation days. This class extends the Employee class by adding specific attributes and methods related to salary calculation, promotions, demotions, and vacation management.

• **Methods**:

• calculate\_salary(): Calculates the monthly salary based on the annual salary.

• promote(new\_annual\_salary, new\_performance\_rating=None): Promotes the employee, increases the salary, optionally updates the performance rating, and prints the promotion details.

• demote(new\_annual\_salary, new\_performance\_rating=None): Demotes the employee, decreases the salary, optionally updates the performance rating, and prints the demotion details.

• adjust\_salary(adjustment): Adjusts the employee’s annual salary and prints the new salary.

• take\_vacation(days): Deducts vacation days from the available balance and prints the remaining days.

• request\_leave\_of\_absence(leave\_days): Requests a leave of absence, adjusts the vacation days, updates the employee’s status to “On Leave,” and prints the leave status.

3.**PartTimeEmployee Class**:

• **Purpose**: The PartTimeEmployee class represents part-time employees who are paid on an hourly basis. This class extends the Employee class and adds methods to manage hourly rates, track hours worked, calculate overtime pay, and adjust hourly wages.

• **Methods**:

• calculate\_salary(): Calculates the total salary, including any overtime pay.

• update\_hours\_worked(additional\_hours): Updates the number of hours worked, tracks overtime, and prints the updated hours and overtime details.

• adjust\_salary(adjustment): Adjusts the hourly rate and prints the new rate.

4.**Contractor Class**:

• **Purpose**: The Contractor class models contractors who work under a fixed-term contract. This class extends the Employee class and includes attributes and methods for managing contract amounts, durations, and renewals.

• **Methods**:

• calculate\_salary(): Calculates the monthly payment based on the contract amount and duration.

• renew\_contract(additional\_months, additional\_amount): Extends the contract duration, increases the contract amount, and prints the details of the contract renewal.

• check\_contract\_status(): Checks if the contract is nearing its end and prints the remaining months or a warning if the end is near.

5.**Intern Class**:

• **Purpose**: The Intern class represents interns who receive a stipend and work for a specified duration. This class extends the Employee class and includes methods to manage the internship duration, assign a mentor, extend the internship, and check the internship status.

• **Methods**:

• calculate\_salary(): Returns the monthly stipend.

• assign\_mentor(mentor\_name): Assigns a mentor to the intern and prints the mentor’s name.

• extend\_internship(additional\_months, additional\_stipend): Extends the internship duration, adjusts the stipend, and prints the extension details.

• check\_internship\_status(): Checks if the internship is nearing its end and prints the remaining months or a warning if the end is near.

6.**display\_all\_employees Function**:

• **Purpose**: The display\_all\_employees() function is used to iterate over a list of employees and display the information for each employee using the display\_info() method.

• **Input**: A list of employee objects.

• **Output**: Prints the details of each employee in the list.