Introduction to Programming

Assignment 8

Objective:

In this assignment, you will implement a simple employee and project management system using Python. The objective is understanding and apply object-oriented programming principles, including inheritance, encapsulation, and composition.

Problem Description:

Create a python program with three classes: Employee, Manager, and Project. The program should allow the creation of employees, assign managers to projects, add team members to projects, and display project details. Follow the requirements given below.

1. **Employee Class**
   - **Attributes:**
     - **Name**: The name of the employee.
     - **Address**: The address of the employee.
     - **Salary**: The employee's salary.
     - **Job title:** The employee's job title.
   - **Methods:**
     - "**generatingReports**": A method that returns a formatted string displaying the employee's name, address, salary, and job title.
2. **Project Class**
   - **Attributes:**
     - **Name**: The project name.
     - **Start date** (type-date): The project start date.
     - **End date** (type-date): The project end date.
     - **Manager** (type- Manager): The manager assigned to the project. This must be an instance of the Manager class.
     - **Team members** (type-list): A list of **Employee instances**(type-Employee) who are project team members.
   - **Methods:**
     - **add_team_member(<Team member>):** Adds a team member to the project. This member must be an instance of Employee.
     - **get_team_members**: Returns a list of all team members assigned to the project with job

title.

- **display_project_details**: Returns a formatted string with project name, start and end dates, manager details, and team members' details.

3. **Manager Class (Inherits from Employee)**
   - **Attributes:**
     - Inherits all attributes from Employee with job_title defaulting to "Manager" and being non-editable.
     - **Reportee** (type-list): Employees who directly reports the manager should be added to this list based on the project the manager oversees.
   - **Methods:**
     - **generatingReports: Overrides Employee's method,** returns all information about manager (name, address, salary, and job title), including the list of team members managed by this manager.
4. **Implementation of class**
   - Create an instance of Manager
   - Create a Project Class and assign the project's manager. (Note: Project should be created with assigned manager but team member can be added later)
   - Create two Employee instances and add them as team members to the project using "**add_ team_member(<Team member>)"** method.
   - Display employee details using the "**generatingReports**" method.
   - Display the project details using the "**display_project_details**" & "**get_team_members"** methods**.**
   - Display the manager's report using "**generatingReports**" method.

Note: *For the Start date & End date (type-date): You can import the "date" from the "datetime library".*

Submission:

- **You must submit a .py file.**
- **Do your own work!** A mark of 0 will be assigned to the entire assignment for work that is not your own and will be handled as per the **Code of Student Rights and Responsibilities**
- **All work must be run and validated to ensure that it is free of errors**. Any assignment that is submitted showing errors that prevent it from running will receive a mark of 0**.**
- **Only apply the knowledge that we have learned in class to this point.** Answers using any syntax or knowledge that we have not covered yet will receive a mark of 0 for that question.
- **Any assignment submitted past the posted due date and time will receive a mark of 0**. Do not wait until the last minute to complete and submit your work.