James Russel E. Adia
CPE21S4

September 23, 2024
CPE 009B

Lab Activity 3 – Inheritance, Encapsulation, and Abstraction

Create a program in python that satisfies the following:

- Inheritance, Encapsulation, and Abstraction concept with ADT list
- Class(Employee: emp_id. emp_name, emp_address, Fulltime: allowance, rate, PartTime: rate)
- Class(Salary: salary_id, Salary, cut_off_date, days_of_work)

Code

Adia - Lab 3 - Inheritance, Encapsulation, Abstraction.py

Lab Act 3 > Adia - Lab 3 - Inheritance, Encapsulation, Abstraction.py > ...

```python
# Assigning the parent class Employee with the attributes emp_id, emp_name, and emp_address
class Employee:
    def __init__(self, emp_id, emp_name, emp_address):
        self.__emp_id = emp_id
        self.__emp_name = emp_name
        self.__emp_address = emp_address

    def get_emp_details(self):
        return {
            "ID": self.__emp_id,
            "Name": self.__emp_name,
            "Address": self.__emp_address
        }

# Creating the child classes FullTime and PartTime that inherits the Employee class
class FullTime(Employee):
    def __init__(self, emp_id, emp_name, emp_address, allowance, rate):
        super().__init__(emp_id, emp_name, emp_address)
        self.__allowance = allowance
        self.__rate = rate

    def calculate_salary(self, days_of_work):
        return (self.__rate * days_of_work) + self.__allowance

class PartTime(Employee):
    def __init__(self, emp_id, emp_name, emp_address, rate):
        super().__init__(emp_id, emp_name, emp_address)
        self.__rate = rate

    def calculate_salary(self, days_of_work):
        return self.__rate * days_of_work
```

```python

# Creating the Salary class that will be used to calculate the salary of the employees
class Salary:
    def __init__(self, salary_id, employee, cut_off_date, days_of_work):
        self.__salary_id = salary_id
        self.__employee = employee
        self.__cut_off_date = cut_off_date
        self.__days_of_work = days_of_work

    def get_salary_details(self):
        salary_amount = self.__employee.calculate_salary(self.__days_of_work)
        return {
            "Salary ID": self.__salary_id,
            "Employee Details": self.__employee.get_emp_details(),
            "Cut-off Date": self.__cut_off_date,
            "Days of Work": self.__days_of_work,
            "Salary Amount": salary_amount
        }

# Example usage
full_time_emp = FullTime(1, "John Doe", "Quezon City", 500, 100)
part_time_emp = PartTime(2, "Jane Smith", "Metro Manila", 80)

full_time_salary = Salary(101, full_time_emp, "09/23/24", 20)
part_time_salary = Salary(102, part_time_emp, "09/22/24", 15)

print(full_time_salary.get_salary_details())
print(part_time_salary.get_salary_details())
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
[Running] python -u "c:\4 - CODING FILES\2ND YEAR\CPE 009B - Object Oriented Programming B\Lab Act 3\Adia - Lab 3 - Inheritance, Encapsulation, Abstraction.py"
{'Salary ID': 101, 'Employee Details': {'ID': 1, 'Name': 'John Doe', 'Address': 'Quezon City'}, 'Cut-off Date': '09/23/24', 'Days of Work': 20, 'Salary Amount': 2500}
{'Salary ID': 102, 'Employee Details': {'ID': 2, 'Name': 'Jane Smith', 'Address': 'Metro Manila'}, 'Cut-off Date': '09/22/24', 'Days of Work': 15, 'Salary Amount': 1200}
```