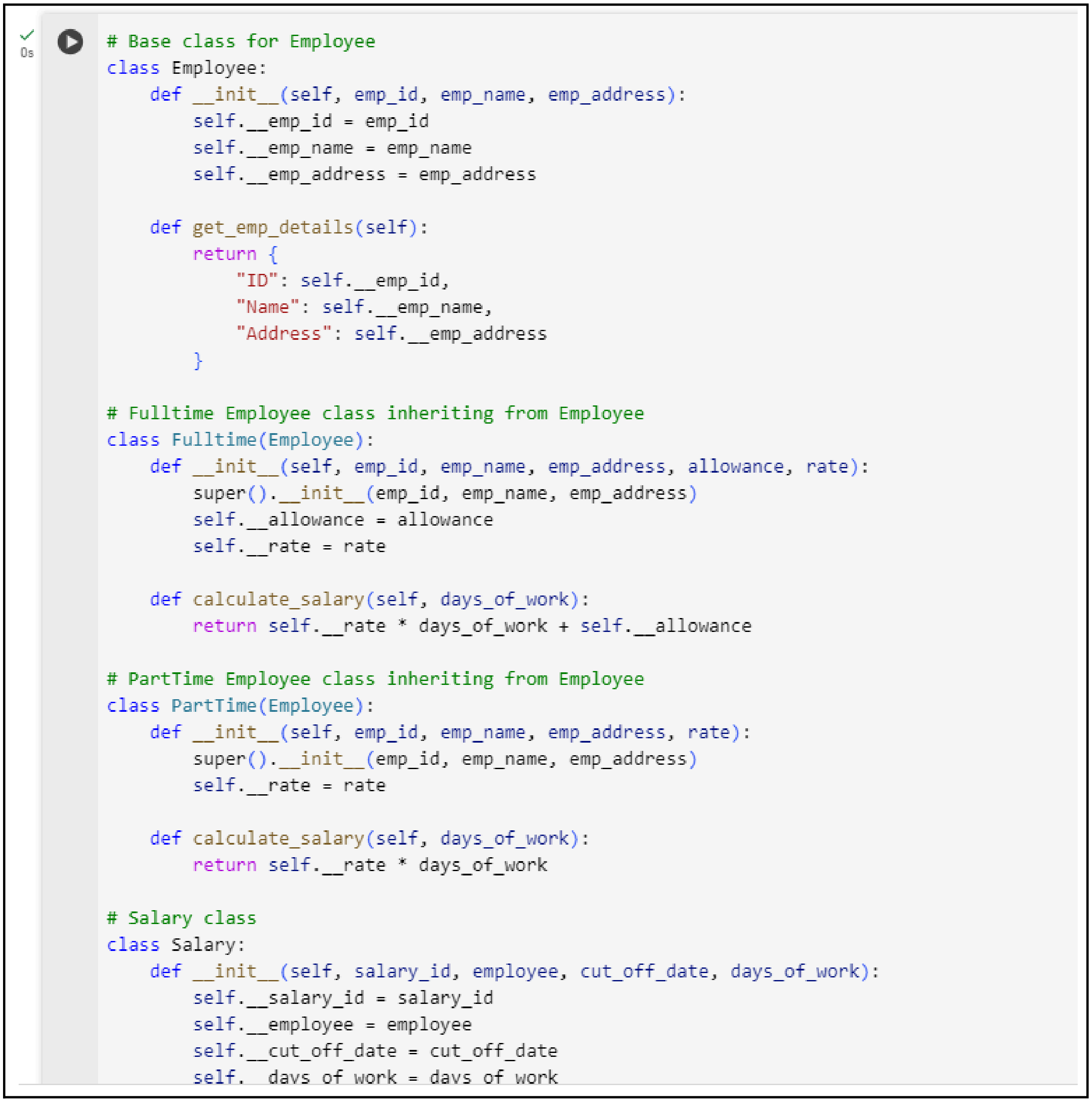
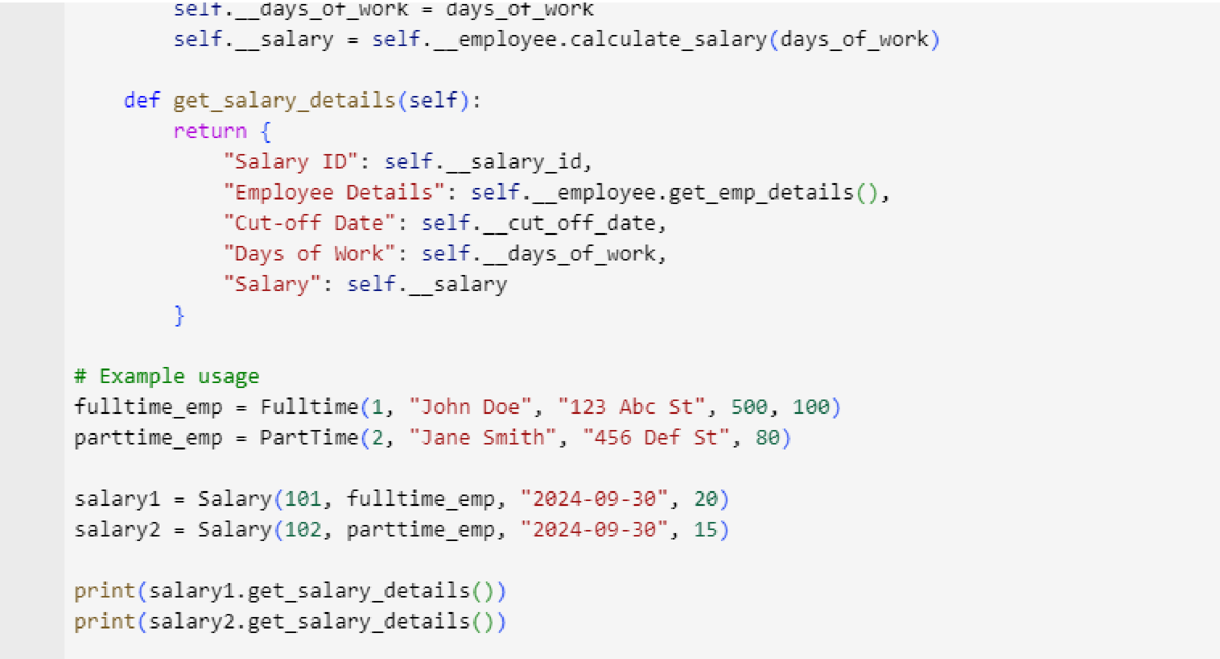
<table>
<tr><th colspan="2">Laboratory Activity 3: Inheritance, Encapsulation, and Abstraction</th></tr>
<tr><td>Rio, Aries, C.</td><td>09/23/24</td></tr>
<tr><td>CPE009B / CPE21S4</td><td>Ma’am Sayo</td></tr>
<tr><td colspan="2"><b>Task:</b></td></tr>
<tr><td colspan="2">Create a program in python that 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)</td></tr>
</table>

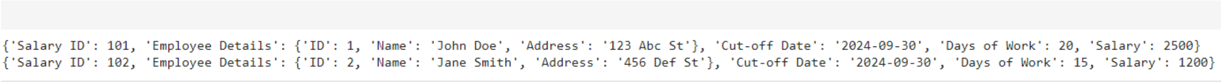


```python
# Base class for Employee
class Employee:
    def __init__(self, emp_id, emp_name, emp_address):
        self.__emp_id = emp_id
        self.__emp_name = emp_name
        self.__emp_address = emp_address

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

# Fulltime Employee class inheriting from Employee
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

# PartTime Employee class inheriting from Employee
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

# Salary class
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
```



```
        self.__days_of_work = days_of_work
        self.__salary = self.__employee.calculate_salary(days_of_work)

    def get_salary_details(self):
        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": self.__salary
        }

# Example usage
fulltime_emp = Fulltime(1, "John Doe", "123 Abc St", 500, 100)
parttime_emp = PartTime(2, "Jane Smith", "456 Def St", 80)

salary1 = Salary(101, fulltime_emp, "2024-09-30", 20)
salary2 = Salary(102, parttime_emp, "2024-09-30", 15)

print(salary1.get_salary_details())
print(salary2.get_salary_details())
```

Output:



```
{'Salary ID': 101, 'Employee Details': {'ID': 1, 'Name': 'John Doe', 'Address': '123 Abc St'}, 'Cut-off Date': '2024-09-30', 'Days of Work': 20, 'Salary': 2500}
{'Salary ID': 102, 'Employee Details': {'ID': 2, 'Name': 'Jane Smith', 'Address': '456 Def St'}, 'Cut-off Date': '2024-09-30', 'Days of Work': 15, 'Salary': 1200}
```