

```

from abc import ABC, abstractmethod
class Employee(ABC):
    def __init__(self, name, employee_id, department):
        self.name = name
        self.employee_id = employee_id
        self.department = department
    @abstractmethod
    def calculate_salary(self):
        pass
    def display_details(self):
        print(f"Name: {self.name}")
        print(f"Employee ID: {self.employee_id}")
        print(f"Department: {self.department}")
class FullTimeEmployee(Employee):
    def __init__(self, name, employee_id, department, monthly_salary):
        super().__init__(name, employee_id, department)
        self.monthly_salary = monthly_salary
    def calculate_salary(self):
        return self.monthly_salary
class PartTimeEmployee(Employee):
    def __init__(self, name, employee_id, department, hourly_wage,
hours_worked):
        super().__init__(name, employee_id, department)
        self.hourly_wage = hourly_wage
        self.hours_worked = hours_worked
    def calculate_salary(self):
        return self.hourly_wage * self.hours_worked
full_time_emp = FullTimeEmployee("Alice", "FT123", "HR", 50000)
part_time_emp = PartTimeEmployee("Bob", "PT456", "IT", 200, 40)
print("Full-Time Employee Details:")
full_time_emp.display_details()
print(f"Salary: {full_time_emp.calculate_salary()}\n")
print("Part-Time Employee Details:")
part_time_emp.display_details()
print(f"Salary: {part_time_emp.calculate_salary()}\n")

```