```
from abc import ABC, abstractmethod
class Employee(ABC):
    def __init__(self,name):
        self.name=name
    @abstractmethod
    def calculate_pay(self):
        pass
class SalariedEmployee(Employee):
    def __init__(self,name,annual_salary):
        super().__init__(name)
        self.annual_salary=annual_salary
    def calculate_pay(self):
        monthly_pay = self.annual_salary/12
        return monthly_pay
class HourlyEmployee(Employee):
    def __init__(self,name,hours_worked,hourly_rate):
        super().__init__(name)
        self.hours_worked =hours_worked
        self.hourly_rate =hourly_rate
    def calculate_pay(self):
        pay = self.hours_worked * self.hourly_rate
        return pay
salaried_employee = SalariedEmployee("John Doe", 60000)
salaried_pay = salaried_employee.calculate_pay()
print(f"Salaried Employee ({salaried_employee.name}) Pay: ${salaried_pay:.2f}")
hourly_employee = HourlyEmployee("Jane Smith", 120, 20)
hourly_pay = hourly_employee.calculate_pay()
print(f"Hourly Employee ({hourly_employee.name}) Pay: ${hourly_pay:.2f}")
```