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):

return self.annual\_salary / 12

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):

return self.hours\_worked \* self.hourly\_rate

def main():

salaried\_employee = SalariedEmployee("John Doe", 60000)

hourly\_employee = HourlyEmployee("Jane Smith", 120, 20)

print(f"Salaried Employee ({salaried\_employee.name}) Pay:

${salaried\_employee.calculate\_pay():.2f}")

print(f"Hourly Employee ({hourly\_employee.name}) Pay: ${hourly\_employee.calculate\_pay():.2f}")

main()