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", "Engineering", 5000)

part\_time\_emp = PartTimeEmployee("Bob", "PT456", "Marketing", 50, 100)

ft\_salary = full\_time\_emp.calculate\_salary()

pt\_salary = part\_time\_emp.calculate\_salary()

print("Full-Time Employee Details:")

full\_time\_emp.display\_details()

print(f"Monthly Salary: {ft\_salary}")

print("\nPart-Time Employee Details:")

part\_time\_emp.display\_details()

print(f"Monthly Salary: {pt\_salary}")