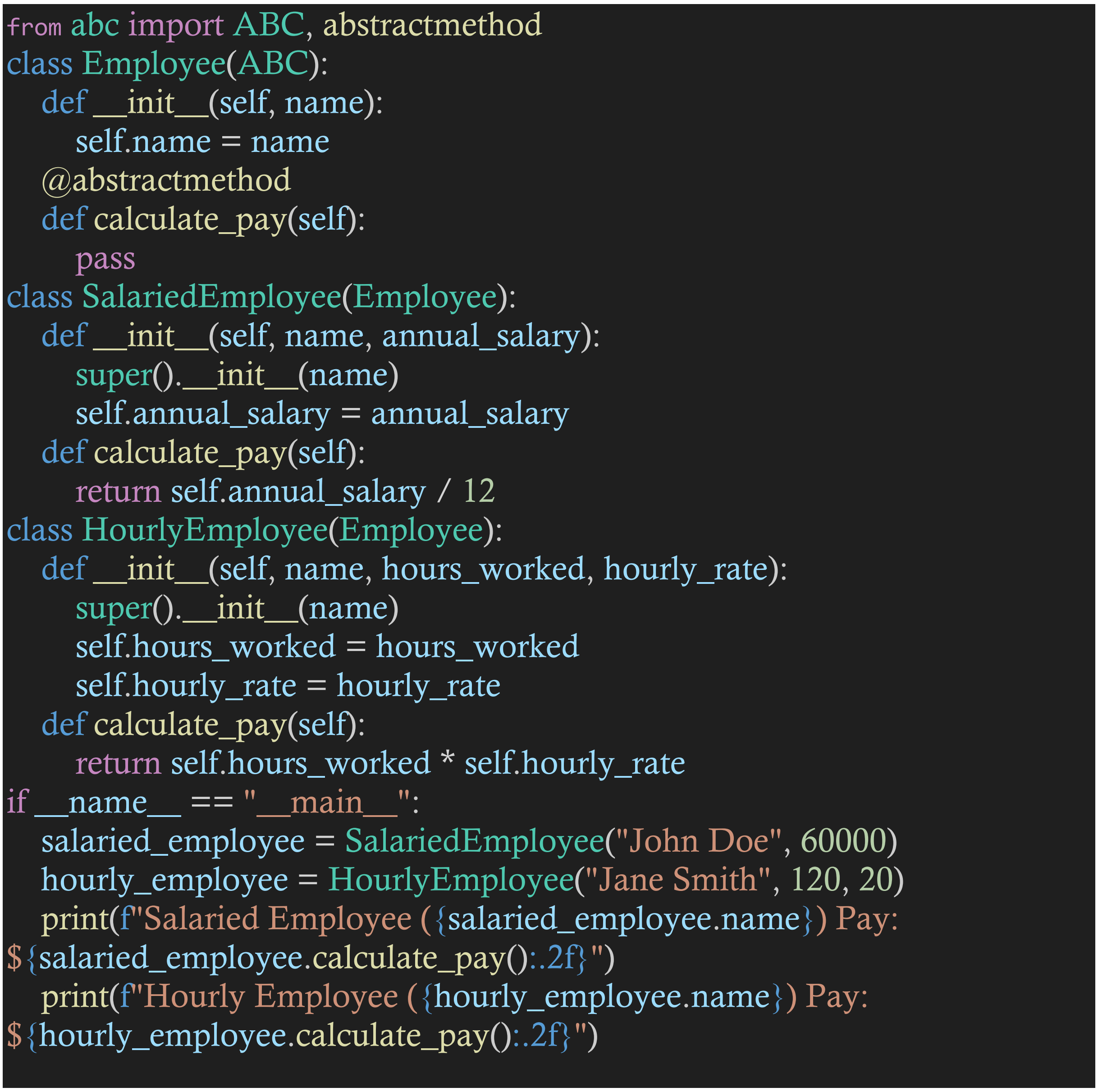

```
from abc import ABC, abstractmethod
class Employee(ABC):
    def __init__(self, name):
        self.name = name
    @abstractmethod
    def calculate_pay(self):
        pass
class SalariedEmployee(Employee):
    def __init__(self, name, annual_salary):
        super().__init__(name)
        self.annual_salary = 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
if __name__ == "__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}")
```