

```

class Employee():
    """
    An employee.
    eid: int        - This employee's ID number
    name: str        - This employee's name
    SIN: int         - This employee's social insurance number
    pay_period: str  - This employee's pay period, either 'weekly' or
                     'monthly'

    This is an abstract class. Only child classes should be instantiated.
    """

    def __init__(self, eid, name, SIN, pay_period):
        """
        (Employee) -> NoneType
        """

        self.eid = eid
        self.name = name
        self.SIN = SIN
        self.pay_period = pay_period

```

```

def amount_of_pay(self):
    """
    (Employee) -> float

    Return the amount that this Employee should be paid in the next
    pay period.
    """

```

```

class HourlyEmployee(Employee):
    """
    An employee whose pay is computed based on an hourly rate.
    hourly_wage: number - This employee's hourly rate of pay
    not_yet_paid: int    - The number of hours of work this Employee
                           has accumulated since their last pay day,
                           and has therefore not yet been paid for.
    """

    def __init__(self, eid, name, SIN, pay_period, hourly_wage):
        """
        (Employee) -> NoneType

        >>> e = HourlyEmployee(23, 'Barney Rubble', 33333333, \
                               'weekly', 1.25)
        >>> e.hourly_wage
        1.25
        >>> e.not_yet_paid
        0
        """

```

```

def amount_of_pay(self):
    """
    (Employee) -> number

    Return the amount that this Employee should be paid in the next
    pay period.

    >>> e = HourlyEmployee(23, 'Barney Rubble', 33333333, \
                           'weekly', 1.25)
    >>> e.log_hours(15)
    >>> e.amount_of_pay()
    18.75
    """

```