Programming Exercise 11-1

```
class Employee:
    def __init__(self, name, id_number):
        self.__name = name
        self.__id_number = id_number

    def set_name(self, name):
        self.__name = name

    def set_id_number(self, id_number):
        self.__id_number = id_number

    def get_name(self):
        return self.__name

    def get_id_number(self):
        return self.__id_number

class ProductionWorker(Employee):
    def __init__(self, name, id_number, shift_number, pay_rate):
        # Call superclass __init__ method.
        Employee.__init__(self, name, id_number)

        # Initialize the shift_number and pay_rate attributes.
        self.__shift_number = shift_number
        self.__pay_rate = pay_rate

    # Mutator functions for shift_number and pay_rate.
    def set_shift_number(self, shift_number):
        self.__shift_number = shift_number

    def set_pay_rate(self, pay_rate):
        self.__pay_rate = pay_rate

    # Accessor functions for shift_number and pay_rate.
    def get_shift_number(self):
        return self.__shift_number

    def get_pay_rate(self):
        return self.__pay_rate
```

```
# Exercise 11-1.py
import emp

def main():
    # Local variables
    worker_name= ''
    worker_id = ''
    worker_shift = 0
    worker_pay = 0.0

    # Get data attributes
    worker_name = input('Enter the name: ')
    worker_id = input('Enter the ID number: ')
    worker_shift = int(input('Enter the shift number: '))
    worker_pay = float(input('Enter the hourly pay rate: '))

    # Create an instance of ProductionWorker
    worker = emp.ProductionWorker(worker_name, worker_id, \
                                  worker_shift, worker_pay)

    # Display information
    print ('Production worker information:')
    print ('Name:', worker.get_name())
    print ('ID number:', worker.get_id_number())
    print ('Shift:', worker.get_shift_number())
    print ('Hourly Pay Rate: $', \
           format(worker.get_pay_rate(), ',.2f'), sep='')

# Call the main function.
main()
```