شایسته گیوه ای

```
class Car():
"""A simple attempt to represent a car."""

def init(self, make, model, year):
        """Initialize attributes to describe a car."""
self.make = make
self.model = model
self.year = year
self.odometer_reading = 0

def get_descriptive_name(self):
           """Return a neatly formatted descriptive name."""
long_name = str(self.year) + ' ' + self.make + ' ' + self.model
return long_name.title()

def read_odometer(self):
             """Print a statement showing the car's mileage."""
Print ("This car has " + str(self.odometer_reading) + " miles on it.")

def update_odometer(self, mileage):
```

```
"""
Set the odometer reading to the given value.
Reject the change if it attempts to roll the odometer back.
"""
if mileage >= self.odometer_reading:
self.odometer_reading = mileage
else:
print ("You can't roll back an odometer!")

def increment_odometer(self, miles):
        """Add the given amount to the odometer reading."""
self.odometer_reading += miles

"""A set of classes used to represent gas and electric cars."""
class Car():
--snip--
class Battery():
"""A simple attempt to model a battery for an electric car."""
def init(self, battery_size=60):
"""Initialize the batteery's attributes."""
self.battery_size = battery_size
def describe_battery(self):
```

```
"""Print a statement describing the battery size."""
print("This car has a " + str(self.battery_size) + "-kWh battery.")
def get_range(self):
"""Print a statement about the range this battery provides."""
if self.battery_size == 70:
range = 240
elif self.battery_size == 85:
range = 270
message = "This car can go approximately " + str(range)
message += " miles on a full charge."
print(message)

class ElectricCar(Car):
"""Models aspects of a car, specific to electric vehicles."""
def init(self, make, model, year):
"""
Initialize attributes of the parent class.
Then initialize attributes specific to an electric car.
"""
super().init(make, model, year)
self.battery = Battery()
```

```
my_beetle = car.Car('volkswagen', 'beetle', 2016)
print(my_beetle.get_descriptive_name())
w my_tesla = car.ElectricCar('tesla', 'roadster', 2016)
print(my_tesla.get_descriptive_name())
  """A set of classes that can be used to represent electric cars."""
 from car import Car
class Battery():
--snip--
    class ElectricCar(Car):
--snip—
 """A class that can be used to represent a car."""
class Car():
--snip--
```