

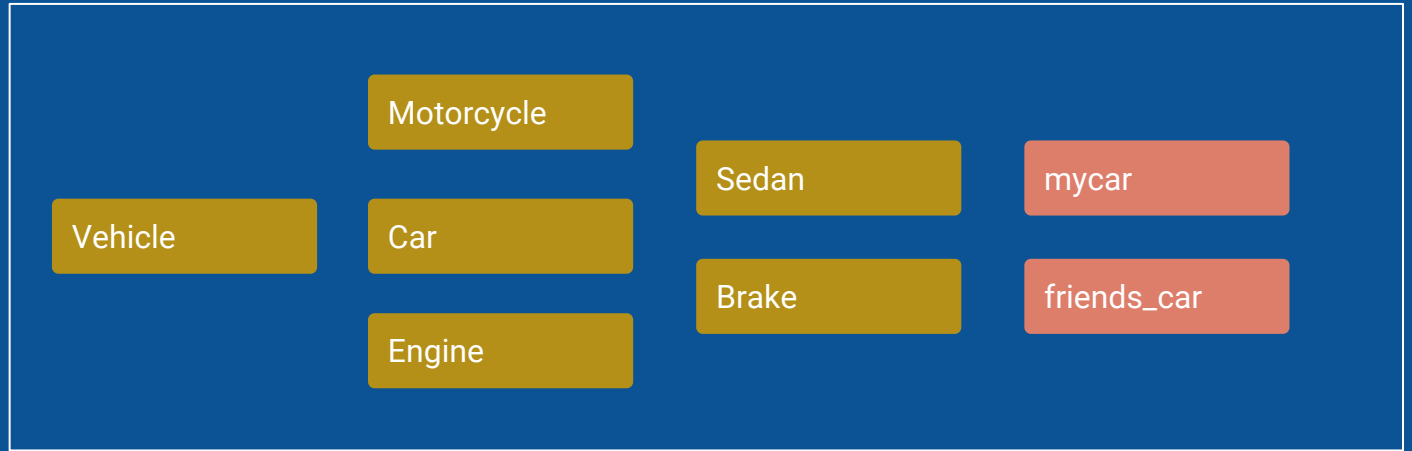
Object Oriented Programming in Python 3

Python 3

Review

- User-defined data types
 - Classes, Objects
- Methods
- Special Methods

Examples of Classes and Objects



Class Relationships

- Classes can be related to each other
- Car *has-a* Engine
- Car *is-a* Vehicle

— has-a

— is-a

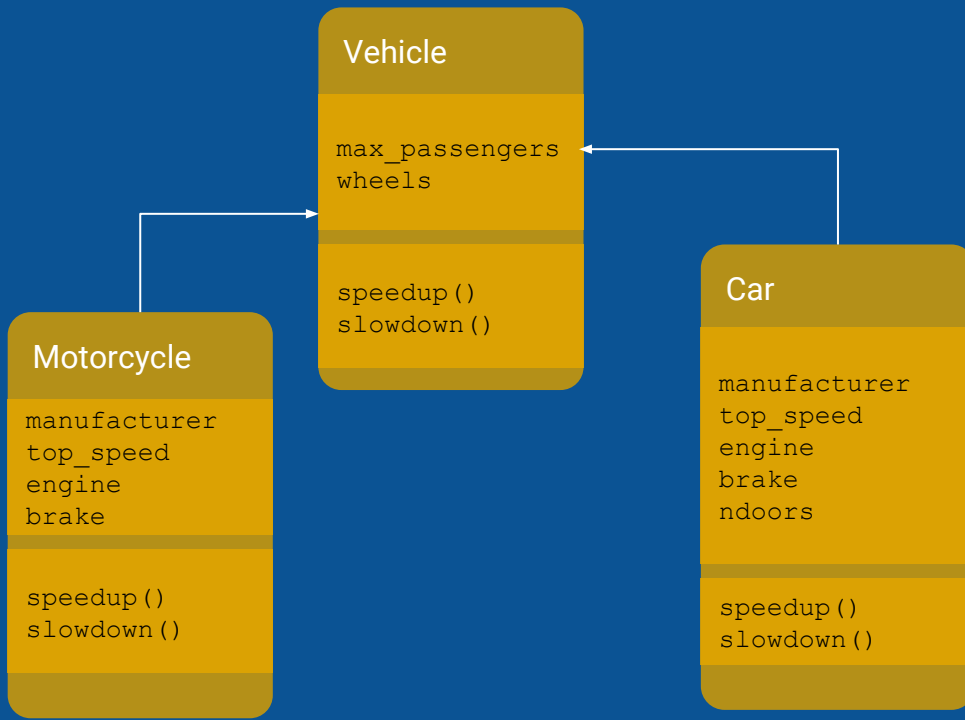
— instance

Vehicle

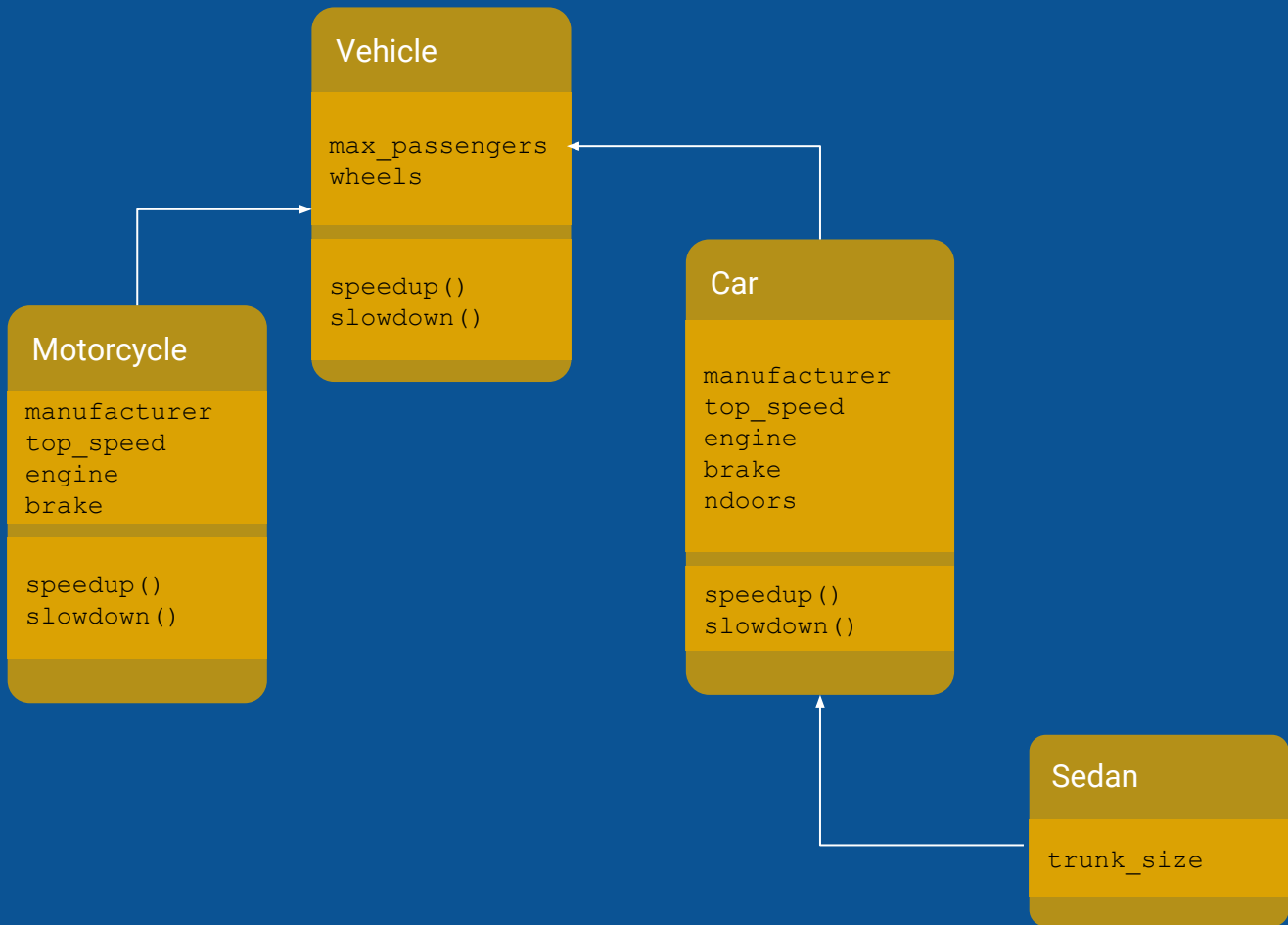
max_passengers
wheels

speedup()
slowdown()

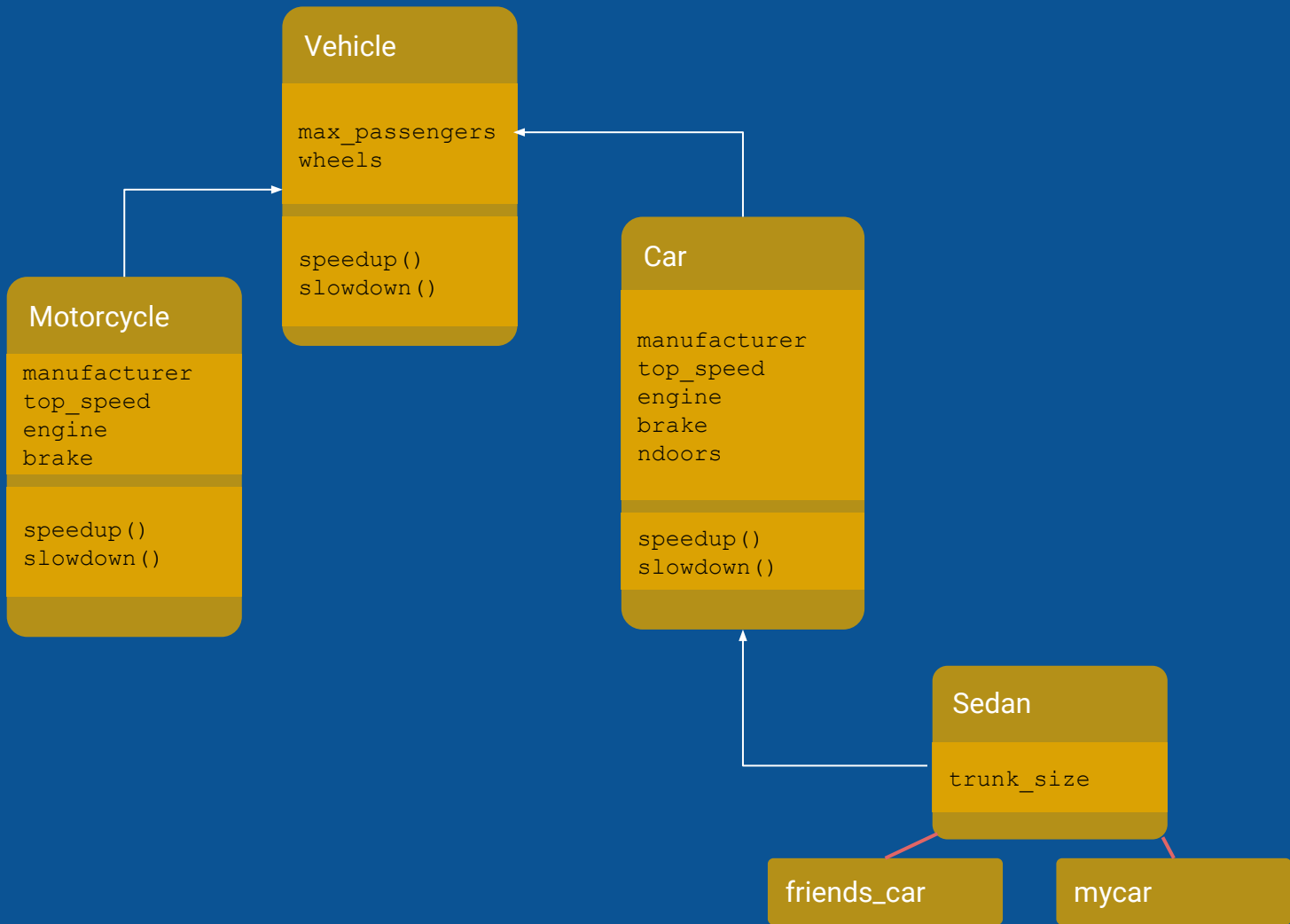
- has-a
- is-a
- instance



- has-a
- is-a
- instance



- has-a
- is-a
- instance



Class Relationships

- Car *is-a* Vehicle
 - Car has all the attributes and methods that Vehicle has
 - With additional attribute `ndoors`
 - `speedup()` and `slowdown()` operate the gas and brake pedals
 - Car is a *subclass* of Vehicle class
 - Vehicle is a *base class* for Car
- Motorcycle is also a Vehicle
 - `speedup()` and `slowdown()` operate the hand controls
 - Motorcycle is a subclass of Vehicle class
 - Vehicle is a base class of Motorcycle

Inheritance

- Is-a relationships are created through inheritance

```
class DerivClass(BaseClass1,...):  
    statement-1  
    statement-2  
    ...
```

- All attributes and methods of BaseClass are inherited by DerivClass

Inheritance

```
class Vehicle:
    def __init__(self, max_pass, wheels):
        self.max_pass = max_pass
        self.wheels = wheels
        self._curspeed = 0

    def speedup(self):
        pass

    def slowdown(self):
        pass

    def printspeed(self):
        print('current speed',self._curspeed,'mph')
```

Inheritance

```
from vehicle import Vehicle

class Car(Vehicle):
    def __init__(self, manuf, top_speed, ndoors):
        super().__init__(max_pass=5, wheels=4)
        self.manuf = manuf
        self.top_speed = top_speed
        self.ndoors = ndoors

    def speedup(self):
        print('speedup by 1 mph. Depress gas pedal')
        if self._curspeed < self.top_speed:
            self._curspeed += 1
        else:
            print('top speed!')

    def slowdown(self):
        print('slowdown by 1 mph. Depress brake pedal')
        if self._curspeed > 0:
            self._curspeed -= 1
```

Inheritance

```
from vehicle import Vehicle

class Motorcycle(Vehicle):
    def __init__(self, manuf, top_speed):
        super().__init__(max_pass=2, wheels=2)
        self.manuf = manuf
        self.top_speed = top_speed

    def speedup(self):
        print('speedup by 1 mph. twist throttle')
        if self._curspeed < self.top_speed:
            self._curspeed += 1
        else:
            print('top speed!')

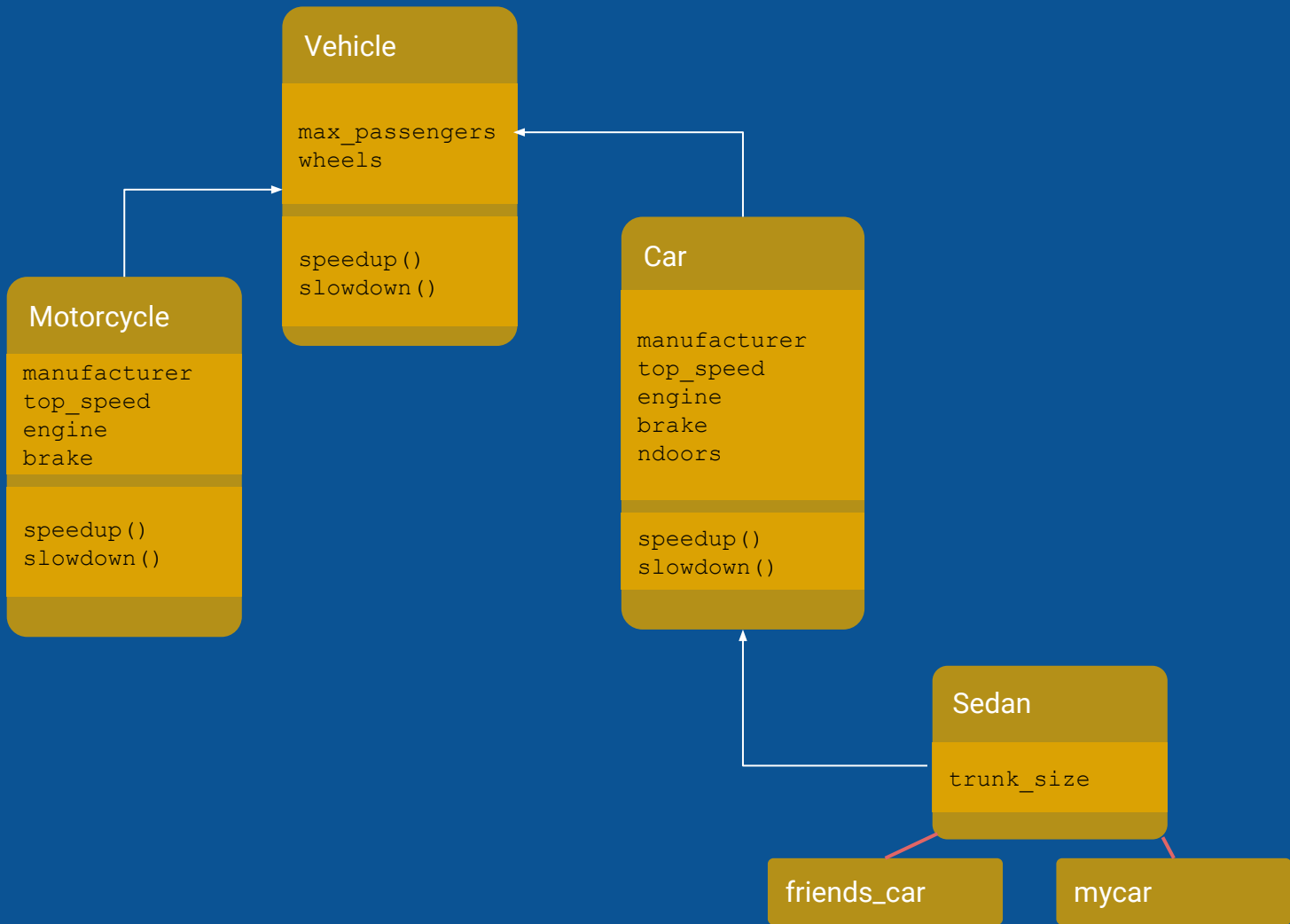
    def slowdown(self):
        print('slowdown by 1 mph. press brake lever')
        if self._curspeed > 0:
            self._curspeed -= 1
```

Inheritance

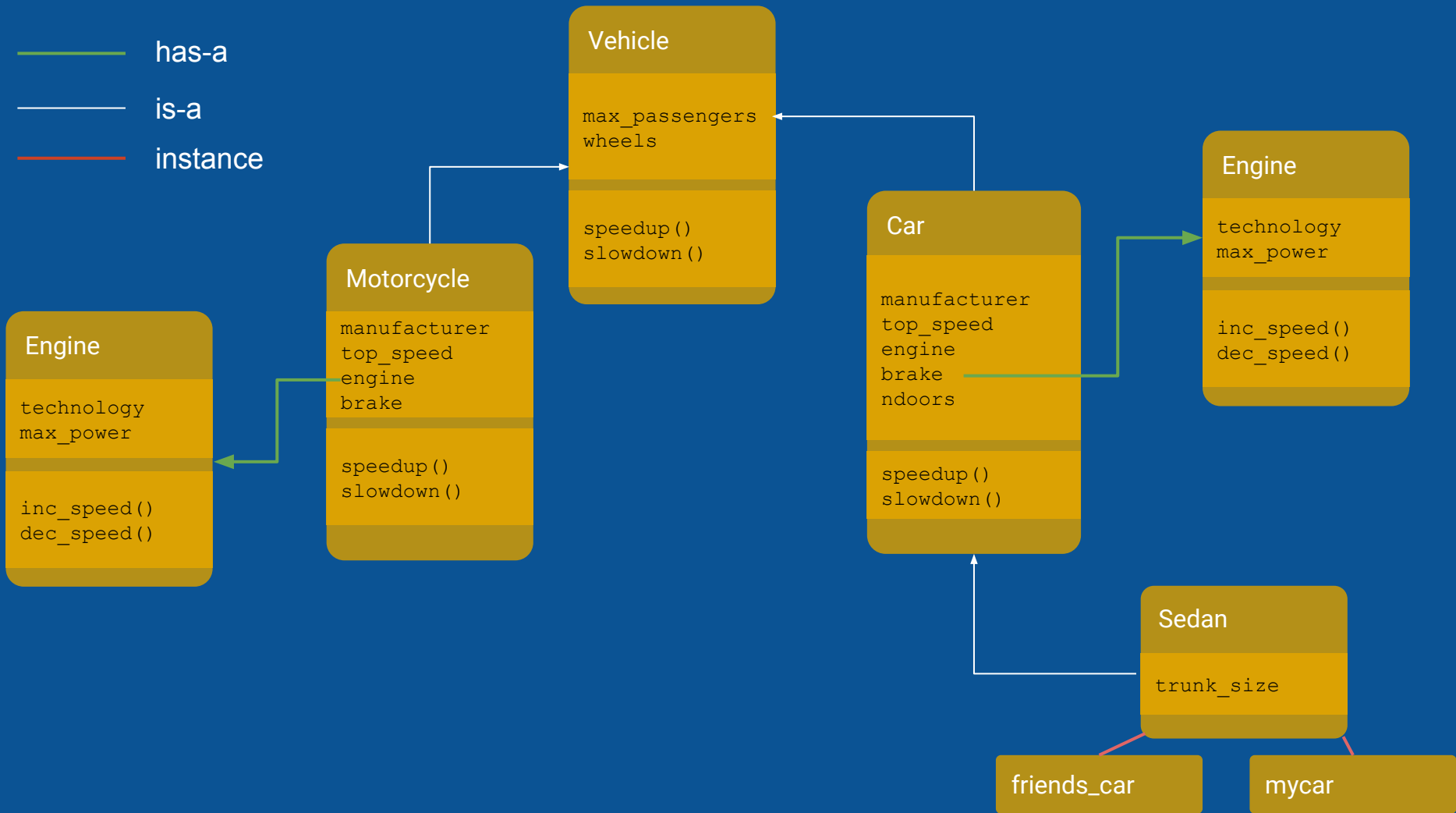
```
class SportsCar(Car):
    def __init__(self,manuf, top_speed):
        super().__init__(manuf, top_speed, 2)
        self.manuf = manuf
        self.top_speed = top_speed

    def speedup(self):
        print('speedup by 5mph')
        self._curspeed += 5
```

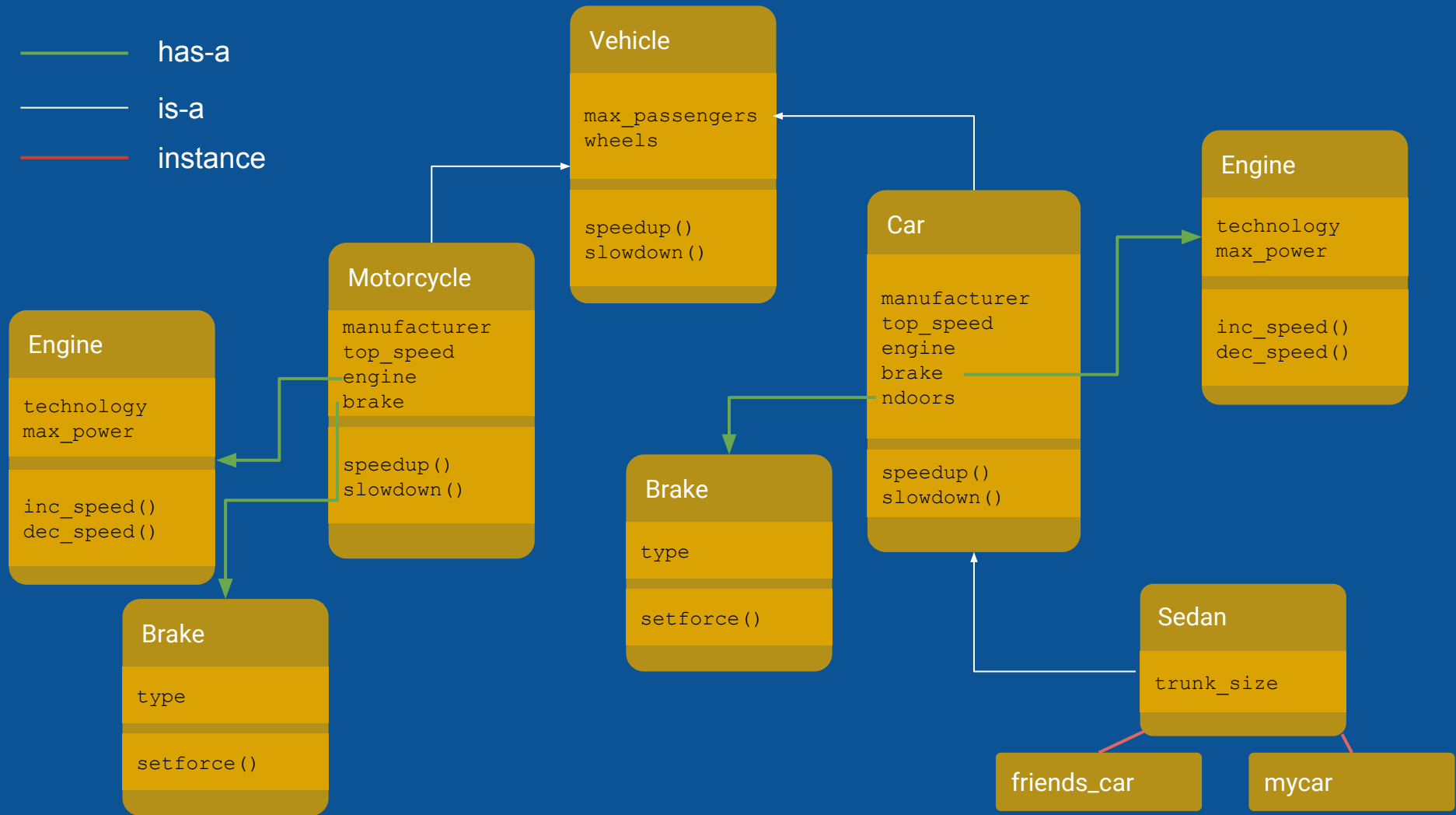
- has-a
- is-a
- instance



- has-a
- is-a
- instance



- has-a
- is-a
- instance



Class Relationships

- Car *has-a* Engine
 - Car has an attribute of type Engine
- Motorcycle also *has-a* Engine
 - Motorcycle has an attribute of type Engine
- Engine can be gas or electric

Composition

- has-a relationship means a class contains an attribute that is of another class/type

Inheritance

```
class Engine:
    def __init__(self, tech='gas'):
        self.tech = tech
        if self.tech == 'gas':
            self._rpm = 500
        elif self.tech == 'electric':
            self.rpm = 0

    def inc_speed(self):
        if self.tech == 'gas':
            print('engine goes vroom...')
        elif self.tech == 'electric':
            print('engine goes sssh...')
        self._rpm += 100

    def dec_speed(self):
        self._rpm -= 100
```

Inheritance

```
class Car(Vehicle):
    def __init__(self, manuf, top_speed, ndoors):
        super().__init__(max_pass=5, wheels=4)
        self.manuf = manuf
        self.top_speed = top_speed
        self.ndoors = ndoors
        self.engine = Engine('gas')

    def speedup(self):
        print('speedup by 1 mph. Depress gas pedal')
        if self._curspeed < self.top_speed:
            self._curspeed += 1
        else:
            print('top speed!')

    def slowdown(self):
        print('slowdown by 1 mph. Depress brake pedal')
        if self._curspeed > 0:
            self._curspeed -= 1
```

Python 3

Managed
Attributes

Properties

```
class C:
    def __init__(self):
        self._x = None

    @property
    def x(self):
        return self._x

    @x.setter
    def x(self, value):
        self._x = value

    @x.deleter
    def x(self):
        del self._x
```


Homework