Tawan Boonma
6310545272

1. **OOP Review**

```
>>> snape = Professor(“Snape")
>>> harry = Student("Harry", snape)
```

Your answer:

There are now 1 students

```
>>> harry.visit_office_hours(snape)
```

Your answer:

Thanks, Snape

```
>>> harry.visit_office_hours(Professor("Hagrid"))
```

Your answer:

Thanks, Hagrid

```
>>> harry.understanding
```

Your answer:

2

```
>>> for name in snape.students:
>>>         print(name)
```

Your answer:

‘Harry’

```
>>> x = Student("Hermione", Professor("McGonagall")).name
```

Your answer:

There are now 2 students

```
>>> x
```

Your answer:

Hermione

```
>>> for name in snape.students:
>>>         print(name)
```

Your answer:

‘Harry’

If we want add more students to Snape’s list, how do you do that?

- Enter snape.add_student(name) ‘name’ is a variable.

2. **Inheritance**

```
class Pet():
    def __init__(self, name, owner):
        self.is_alive = True    # It's alive!!!
        self.name = name
        self.owner = owner
    def eat(self, thing):
        print(self.name + " ate a " + str(thing) + "!")
    def talk(self):
        print(self.name)

class Dog(Pet):
    def talk(self):
        print(self.name + ' says woof!')
```

```
class Cat(Pet):
    def __init__(self, name, owner, lives=9):
        super().__init__(name, owner)
        self.lives = lives

    def talk(self):
    """ Print out a cat's greeting.

    """
        print(self.name + “ says meow!”)
    def lose_life(self):
    """Decrements a cat's life by 1. When lives reaches zero, 'is_alive'
    becomes False. If this is called after lives has reached zero, print out
    that the cat has no more lives to lose.
    """

        if not self.is_alive:

            print(“The cat has no more lives to lose.”)

        else:

            self.lives -= 1

            if self.lives == 0:

                self.is_alive = False


>>> Cat('Thomas', 'Tammy').talk()

Thomas says meow!


class NoisyCat(Cat):
    """A Cat that repeats things twice."""


    def talk(self):
    """Talks twice as much as a regular cat."""

        super().talk()

        super().talk()


>>> NoisyCat('Magic', 'James').talk()

Magic says meow!
Magic says meow!
```

3. More inheritance

```
>>> deneros_car = Car('Tesla', 'Model S')
>>> deneros_car.model
```

Your answer:

‘Model S’

```
>>> deneros_car.gas = 10
>>> deneros_car.drive()
```

Your answer:

‘Tesla Model S goes vroom!’

```
>>> deneros_car.drive()
```

Your answer:

‘Cannot drive!’

```
>>> deneros_car.fill_gas()
```

Your answer:

‘Gas level: 20’

```
>>> deneros_car.gas
```

Your answer:

20

```
>>> Car.gas
```

Your answer:

30

```
>>> deneros_car = Car('Tesla', 'Model S')
>>> deneros_car.wheels = 2
>>> deneros_car.wheels
```

Your answer:

2

```
>>> Car.num_wheels
```

Your answer:

4

```
>>> deneros_car.drive()
```

Your answer:

‘Cannot drive!’

```
>>> Car.drive()
```

Your answer:

TypeError: drive() missing 1 required positional argument: 'self'

```
>>> Car.drive(deneros_car)
```

Your answer:

‘Cannot drive!’