Name: Steve Hommy

Pair: -

Amount of completed tasks: 10

Which tasks were left undone or incomplete: 0

Self-assessment:

This exercise was easy for me because I have worked with classes and inheritance before. Doing this exercise, I learned how to print super class in inherited class. I understood everything that was going on

Test report

Write the test report yourself to each coding task (task number, input/action, desired output and then the testing evidence (actual output)). Add rows if necessary. Include answers to theoretical questions and pseudocode to this return document as well in addition to code screen captures. Actual output can be a screen capture of the terminal showing the output.

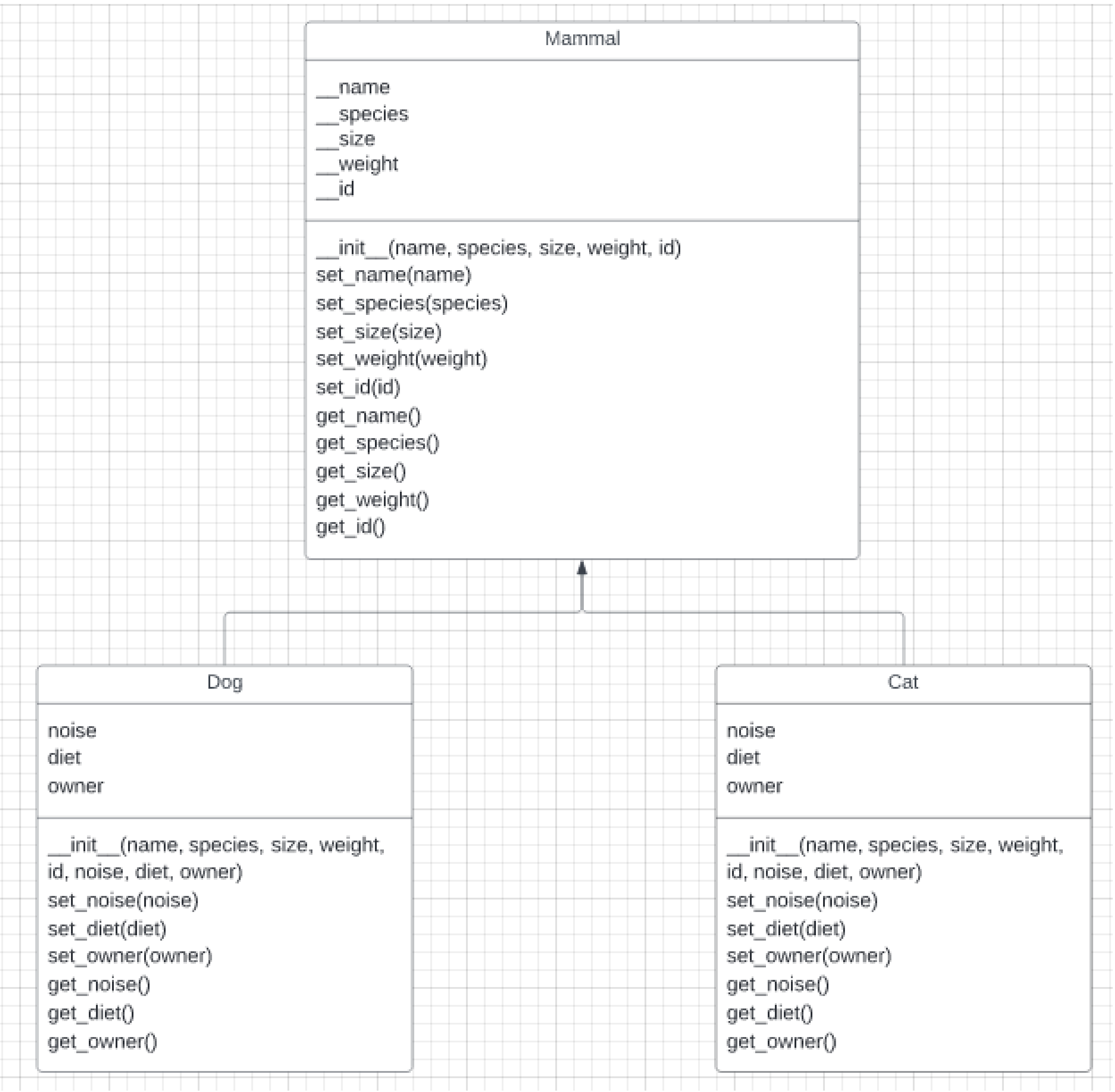
Task	Input / action	Desired output	Actual output (use red color if desired output != actual output)
2	<Run Program>	Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve Name: Snuf Species: Dog Size: 40 Weight: 20 ID: 2 Noise of the animal: Bark Diet: Feed 3 times a day Owner: Joe	Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve Name: Snuf Species: Dog Size: 40 Weight: 20 ID: 2 Noise of the animal: Bark Diet: Feed 3 times a day Owner: Joe
4	<Run Program>	Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve Color: black Name: Snuf Species: Cat Size: 20 Weight: 10	Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve Color: black Name: Snuf Species: Cat Size: 20 Weight: 10

		ID: 2 Noise of the animal: Miaw Diet: Feed 1 times a day Owner: Steve Color: brown Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe Speed: 40 Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe Speed: 50	ID: 2 Noise of the animal: Miaw Diet: Feed 1 times a day Owner: Steve Color: brown Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe Speed: 40 Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe Speed: 50
6	<Run Program>	Name: Steve Gender: Male Class Number: 3 Student number: 324198 Year of study: 3 Name: Sanna Gender: Female Class Number: 3 Academic rank: Lecturer Phone number: 358954381273	Name: Steve Gender: Male Class Number: 3 Student number: 324198 Year of study: 3 Name: Sanna Gender: Female Class Number: 3 Academic rank: Lecturer Phone number: 358954381273
7	<Run Program>	Name: Steve Gender: Male Class Number: 3 Student number: 324198 Year of study: 3 domestic animal is: Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve and wild animal:	Name: Steve Gender: Male Class Number: 3 Student number: 324198 Year of study: 3 domestic animal is: Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve and wild animal:

		Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe Name: Sanna Gender: Female Class Number: 3 Academic rank: Lecturer Phone number: 358954381273 domestic animal is: Name: Snuf Species: Cat Size: 20 Weight: 10 ID: 2 Noise of the animal: Miaw Diet: Feed 1 times a day Owner: Sanna and wild animal: Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe	Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe Name: Sanna Gender: Female Class Number: 3 Academic rank: Lecturer Phone number: 358954381273 domestic animal is: Name: Snuf Species: Cat Size: 20 Weight: 10 ID: 2 Noise of the animal: Miaw Diet: Feed 1 times a day Owner: Sanna and wild animal: Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe
8	<Run Program>	Name: Steve Gender: Male Class Number: 3 Student number: 324198 Year of study: 3 domestic animals are: Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve	Name: Steve Gender: Male Class Number: 3 Student number: 324198 Year of study: 3 domestic animals are: Name: Bob Species: Dog Size: 60 Weight: 30 ID: 1 Noise of the animal: Bark Diet: Feed 2 times a day Owner: Steve

		Name: Pier Species: Hamster Size: 2 Weight: 5 ID: 3 Noise of the animal: Squeek Diet: Feed 1 times a day Owner: Steve and wild animals are: Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe Name: Sanna Gender: Female Class Number: 3 Academic rank: Lecturer Phone number: 358954381273 domestic animals are: Name: Snuf Species: Cat Size: 20 Weight: 10 ID: 2 Noise of the animal: Miaw Diet: Feed 1 times a day Owner: Sanna Name: Nown Species: Rabbit Size: 5 Weight: 10 ID: 4 Noise of the animal: Shhh	Name: Pier Species: Hamster Size: 2 Weight: 5 ID: 3 Noise of the animal: Squeek Diet: Feed 1 times a day Owner: Steve and wild animals are: Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe Name: Sanna Gender: Female Class Number: 3 Academic rank: Lecturer Phone number: 358954381273 domestic animals are: Name: Snuf Species: Cat Size: 20 Weight: 10 ID: 2 Noise of the animal: Miaw Diet: Feed 1 times a day Owner: Sanna Name: Nown Species: Rabbit

		Diet: Feed 1.5 times a day Owner: Sanna and wild animals are: Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe	Size: 5 Weight: 10 ID: 4 Noise of the animal: Shhh Diet: Feed 1.5 times a day Owner: Sanna and wild animals are: Name: Max Species: Lion Size: 50 Weight: 80 ID: 4 Noise of the animal: Grawr Diet: Feed 4 times a day Zoo keeper: Joe Name: Nie Species: Tiger Size: 40 Weight: 60 ID: 3 Noise of the animal: Rawr Diet: Feed 3 times a day Zoo keeper: Joe
10	<Run Program>	Our first car is: Brand: Honda Tyre: Continental Body style: Hatchback 0 to 100 in: 8.5 seconds Engine size: 1.6l Tank size: 100l Our second car is: Brand: Tesla Tyre: Nokia Body style: Sedan 0 to 100 in: 4.5 seconds Electric power: 250W Battery size: 1000 000A Honda will reach 0 to 100 in 8.5 seconds Tesla will reach 0 to 100 in 4.5 seconds	Our first car is: Brand: Honda Tyre: Continental Body style: Hatchback 0 to 100 in: 8.5 seconds Engine size: 1.6l Tank size: 100l Our second car is: Brand: Tesla Tyre: Nokia Body style: Sedan 0 to 100 in: 4.5 seconds Electric power: 250W Battery size: 1000 000A Honda will reach 0 to 100 in 8.5 seconds Tesla will reach 0 to 100 in 4.5 seconds



Mammal

__name
__species
__size
__weight
__id

__init__(name, species, size, weight, id)
set_name(name)
set_species(species)
set_size(size)
set_weight(weight)
set_id(id)
get_name()
get_species()
get_size()
get_weight()
get_id()

Dog

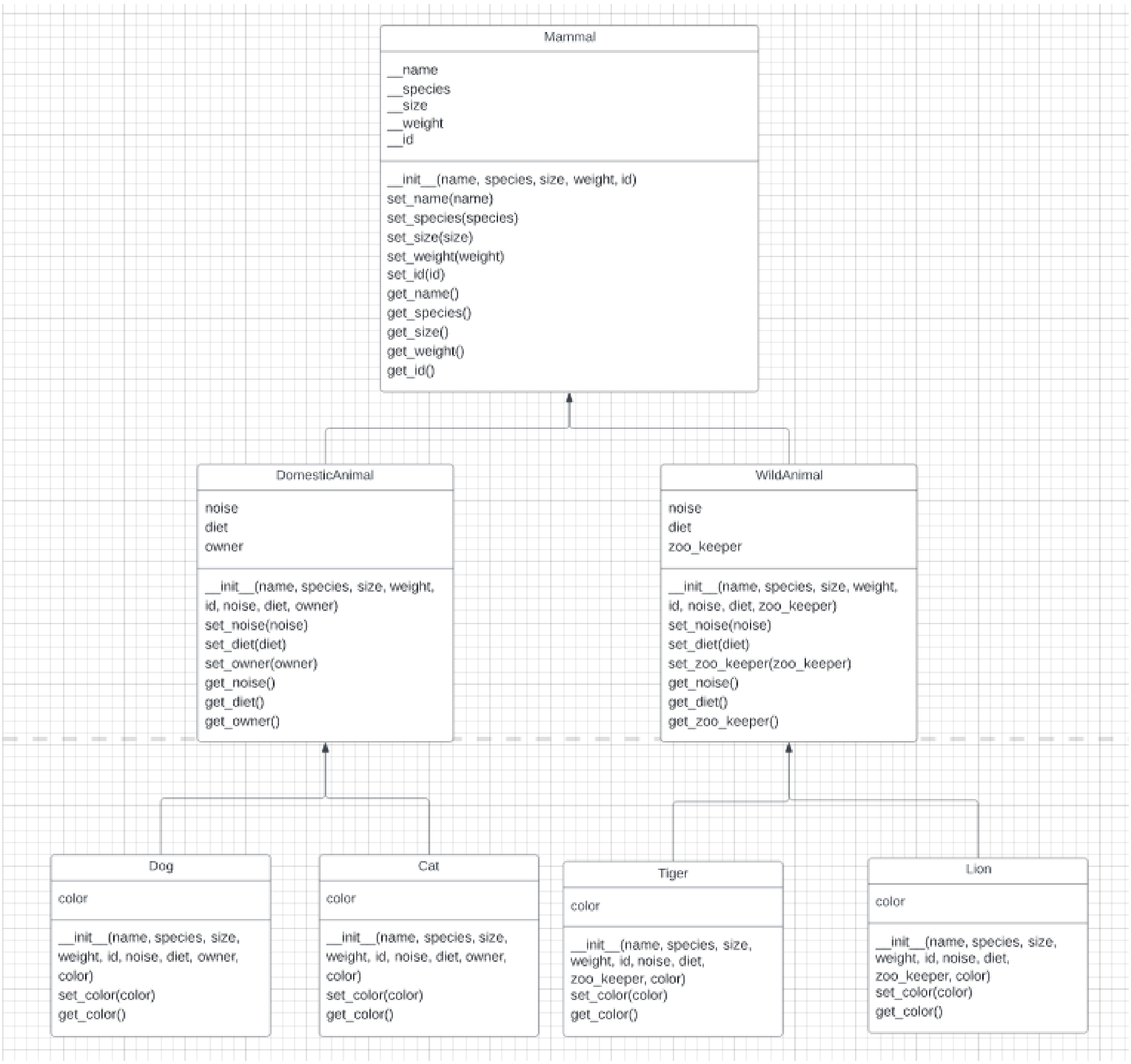
noise
diet
owner

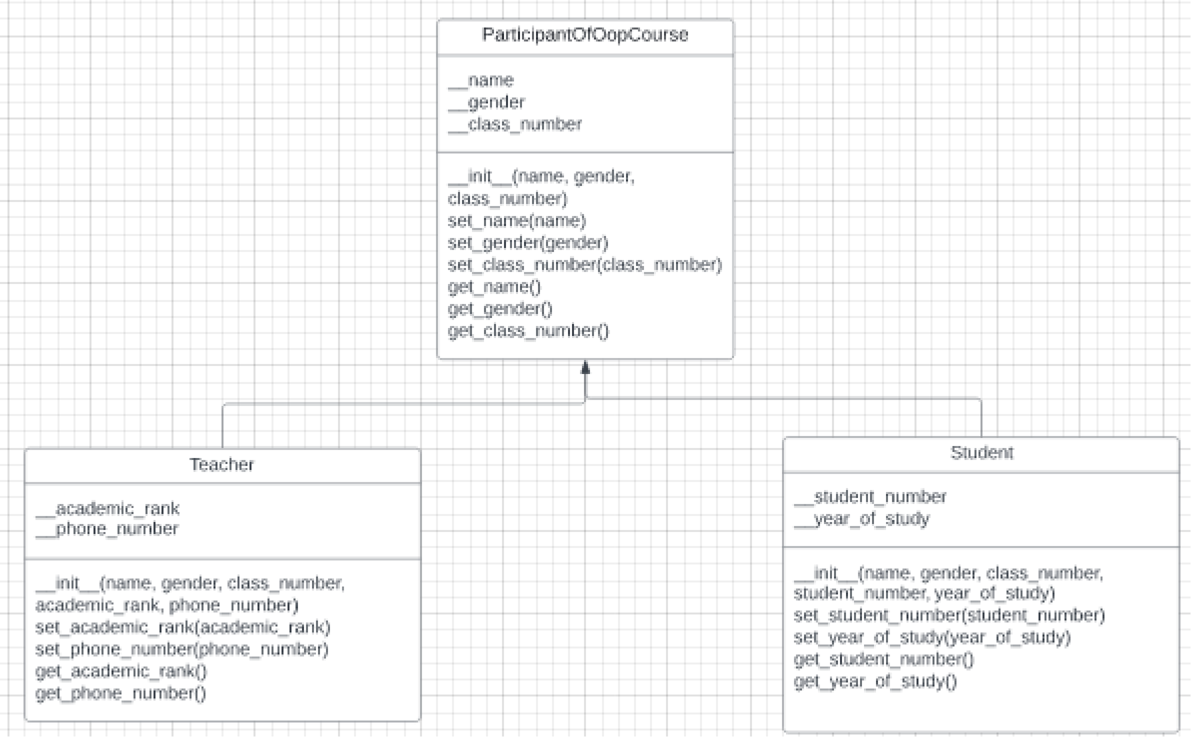
__init__(name, species, size, weight, id, noise, diet, owner)
set_noise(noise)
set_diet(diet)
set_owner(owner)
get_noise()
get_diet()
get_owner()

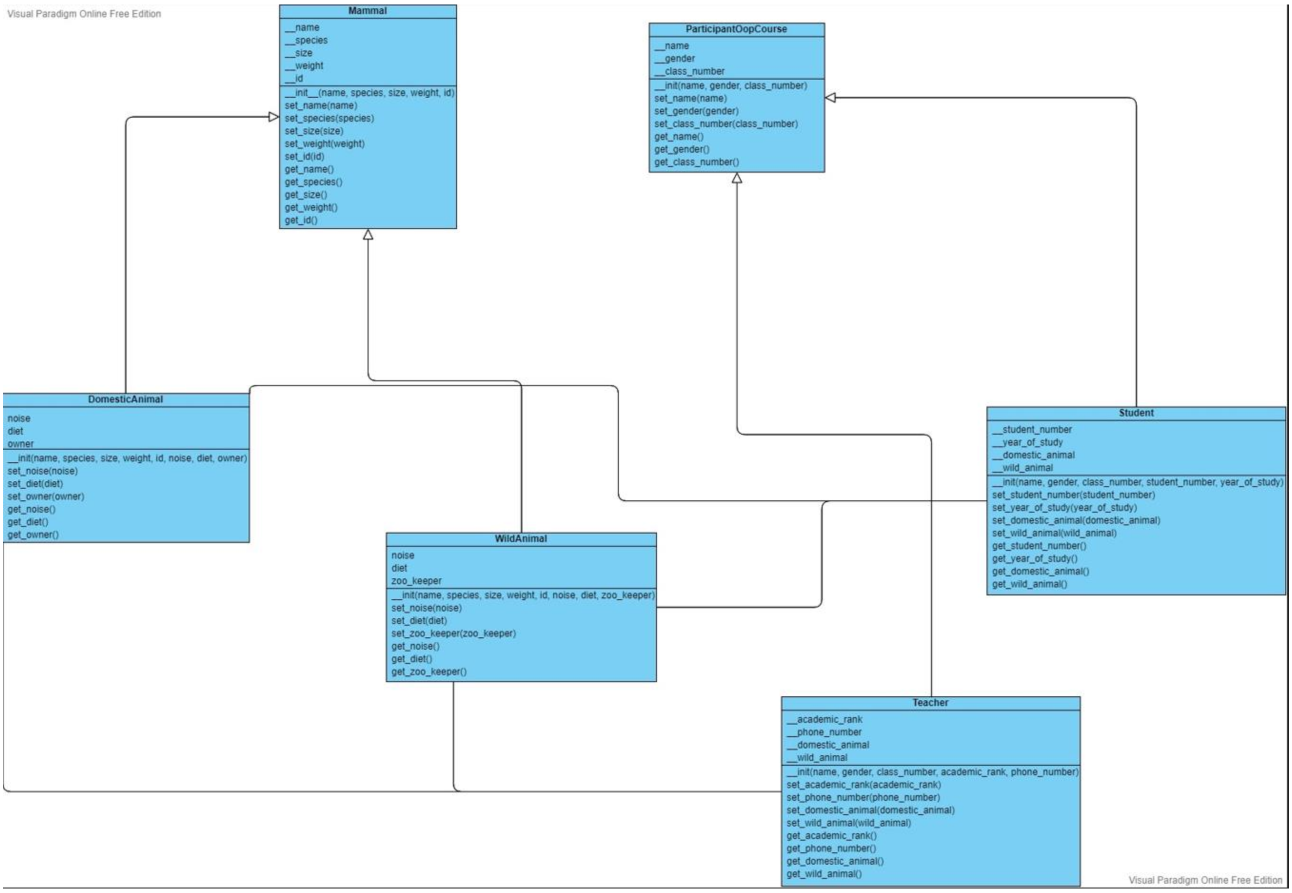
Cat

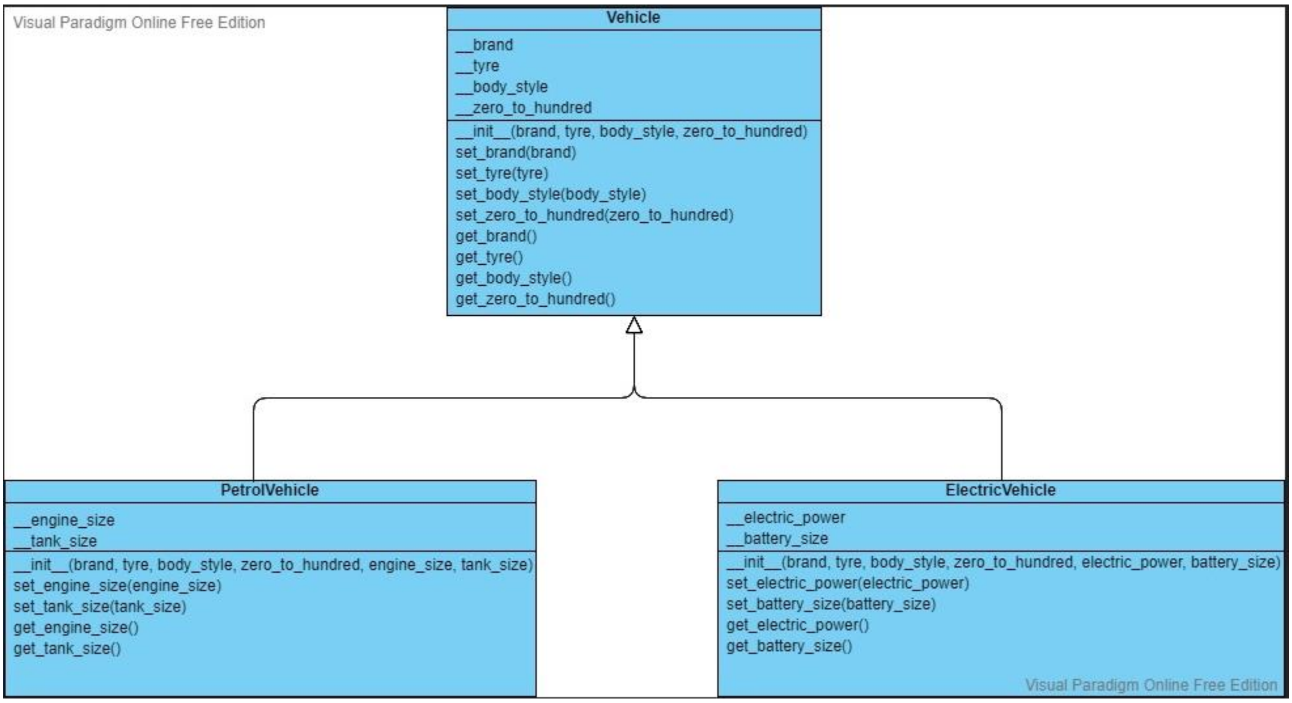
noise
diet
owner

__init__(name, species, size, weight, id, noise, diet, owner)
set_noise(noise)
set_diet(diet)
set_owner(owner)
get_noise()
get_diet()
get_owner()

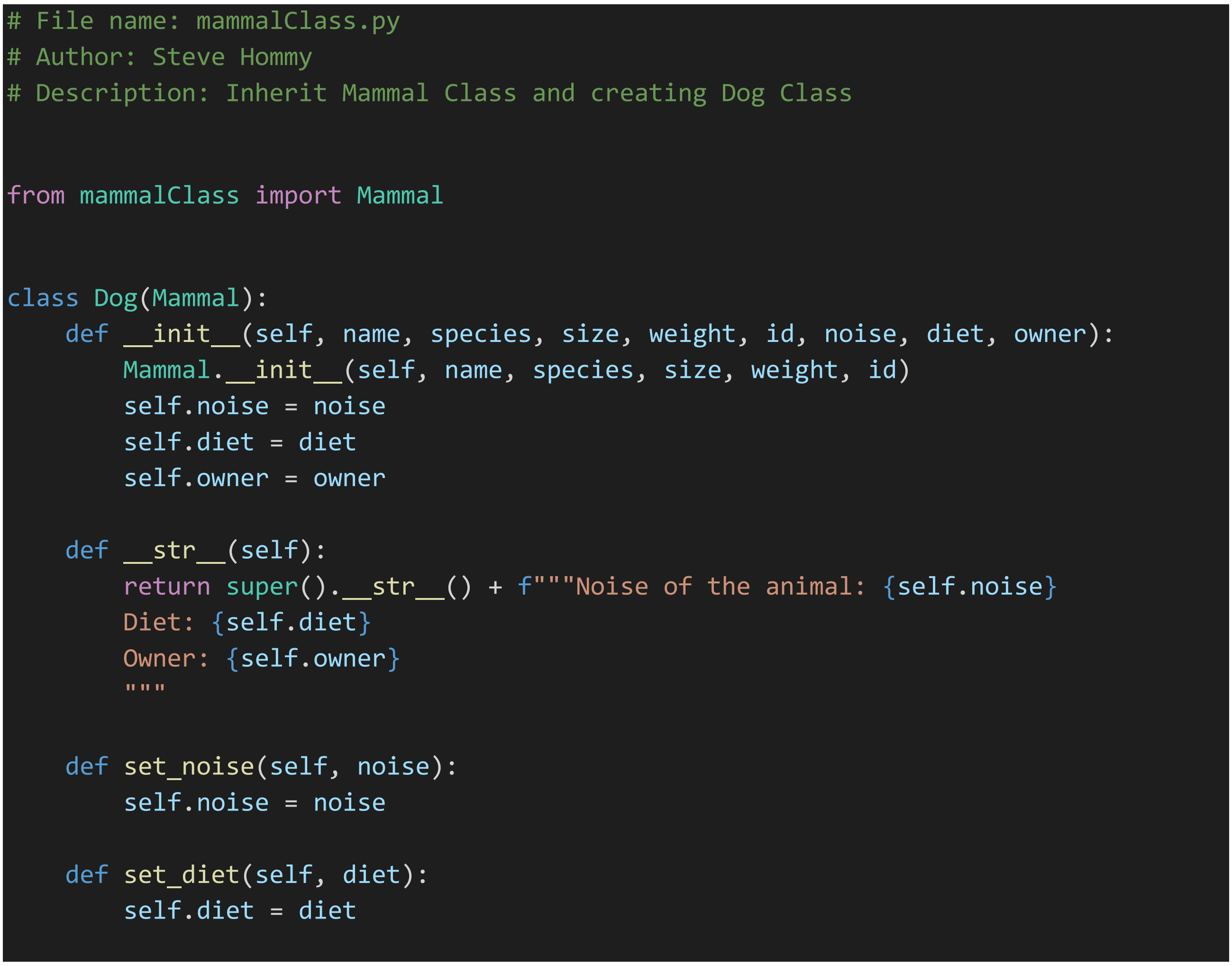

Mammal
__name
__species
__size
__weight
__id
__init__(name, species, size, weight, id)
set_name(name)
set_species(species)
set_size(size)
set_weight(weight)
set_id(id)
get_name()
get_species()
get_size()
get_weight()
get_id()
DomesticAnimal
noise
diet
owner
__init__(name, species, size, weight, id, noise, diet, owner)
set_noise(noise)
set_diet(diet)
set_owner(owner)
get_noise()
get_diet()
get_owner()
WildAnimal
noise
diet
zoo_keeper
__init__(name, species, size, weight, id, noise, diet, zoo_keeper)
set_noise(noise)
set_diet(diet)
set_zoo_keeper(zoo_keeper)
get_noise()
get_diet()
get_zoo_keeper()
Dog
color
__init__(name, species, size, weight, id, noise, diet, owner, color)
set_color(color)
get_color()
Cat
color
__init__(name, species, size, weight, id, noise, diet, owner, color)
set_color(color)
get_color()
Tiger
color
__init__(name, species, size, weight, id, noise, diet, zoo_keeper, color)
set_color(color)
get_color()
Lion
color
__init__(name, species, size, weight, id, noise, diet, zoo_keeper, color)
set_color(color)
get_color()


ParticipantOfOopCourse
__name
__gender
__class_number
__init__(name, gender, class_number)
set_name(name)
set_gender(gender)
set_class_number(class_number)
get_name()
get_gender()
get_class_number()
Teacher
__academic_rank
__phone_number
__init__(name, gender, class_number, academic_rank, phone_number)
set_academic_rank(academic_rank)
set_phone_number(phone_number)
get_academic_rank()
get_phone_number()
Student
__student_number
__year_of_study
__init__(name, gender, class_number, student_number, year_of_study)
set_student_number(student_number)
set_year_of_study(year_of_study)
get_student_number()
get_year_of_study()

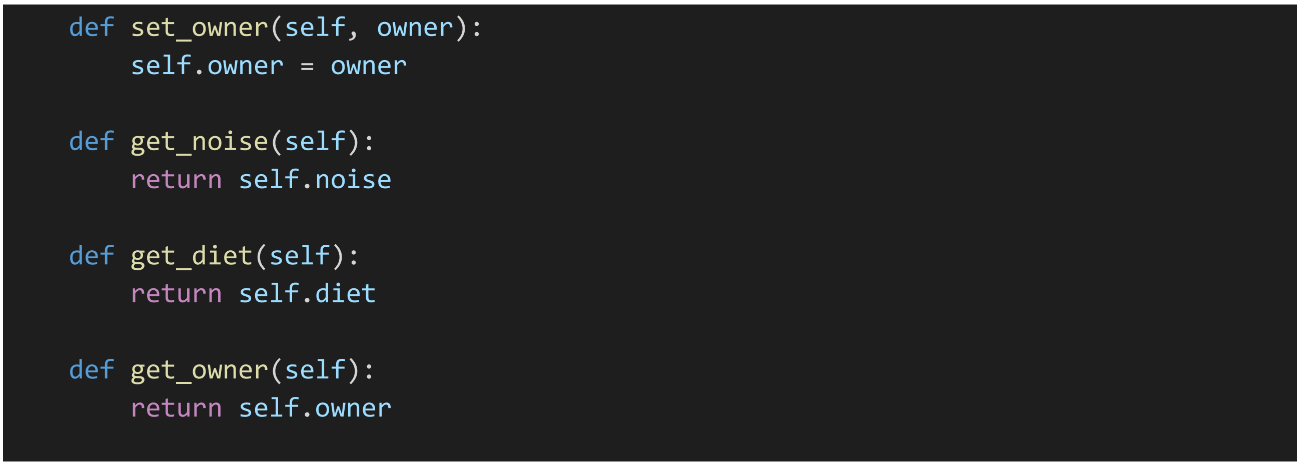

Visual Paradigm Online Free Edition
Mammal
__name
__species
__size
__weight
__id
__init__(name, species, size, weight, id)
set_name(name)
set_species(species)
set_size(size)
set_weight(weight)
set_id(id)
get_name()
get_species()
get_size()
get_weight()
get_id()
ParticipantOopCourse
__name
__gender
__class_number
__init(name, gender, class_number)
set_name(name)
set_gender(gender)
set_class_number(class_number)
get_name()
get_gender()
get_class_number()
DomesticAnimal
noise
diet
owner
__init(name, species, size, weight, id, noise, diet, owner)
set_noise(noise)
set_diet(diet)
set_owner(owner)
get_noise()
get_diet()
get_owner()
WildAnimal
noise
diet
zoo_keeper
__init(name, species, size, weight, id, noise, diet, zoo_keeper)
set_noise(noise)
set_diet(diet)
set_zoo_keeper(zoo_keeper)
get_noise()
get_diet()
get_zoo_keeper()
Student
__student_number
__year_of_study
__domestic_animal
__wild_animal
__init(name, gender, class_number, student_number, year_of_study)
set_student_number(student_number)
set_year_of_study(year_of_study)
set_domestic_animal(domestic_animal)
set_wild_animal(wild_animal)
get_student_number()
get_year_of_study()
get_domestic_animal()
get_wild_animal()
Teacher
__academic_rank
__phone_number
__domestic_animal
__wild_animal
__init(name, gender, class_number, academic_rank, phone_number)
set_academic_rank(academic_rank)
set_phone_number(phone_number)
set_domestic_animal(domestic_animal)
set_wild_animal(wild_animal)
get_academic_rank()
get_phone_number()
get_domestic_animal()
get_wild_animal()
Visual Paradigm Online Free Edition


Visual Paradigm Online Free Edition
Vehicle
__brand
__tyre
__body_style
__zero_to_hundred
__init__(brand, tyre, body_style, zero_to_hundred)
set_brand(brand)
set_tyre(tyre)
set_body_style(body_style)
set_zero_to_hundred(zero_to_hundred)
get_brand()
get_tyre()
get_body_style()
get_zero_to_hundred()
PetrolVehicle
__engine_size
__tank_size
__init__(brand, tyre, body_style, zero_to_hundred, engine_size, tank_size)
set_engine_size(engine_size)
set_tank_size(tank_size)
get_engine_size()
get_tank_size()
ElectricVehicle
__electric_power
__battery_size
__init__(brand, tyre, body_style, zero_to_hundred, electric_power, battery_size)
set_electric_power(electric_power)
set_battery_size(battery_size)
get_electric_power()
get_battery_size()
Visual Paradigm Online Free Edition

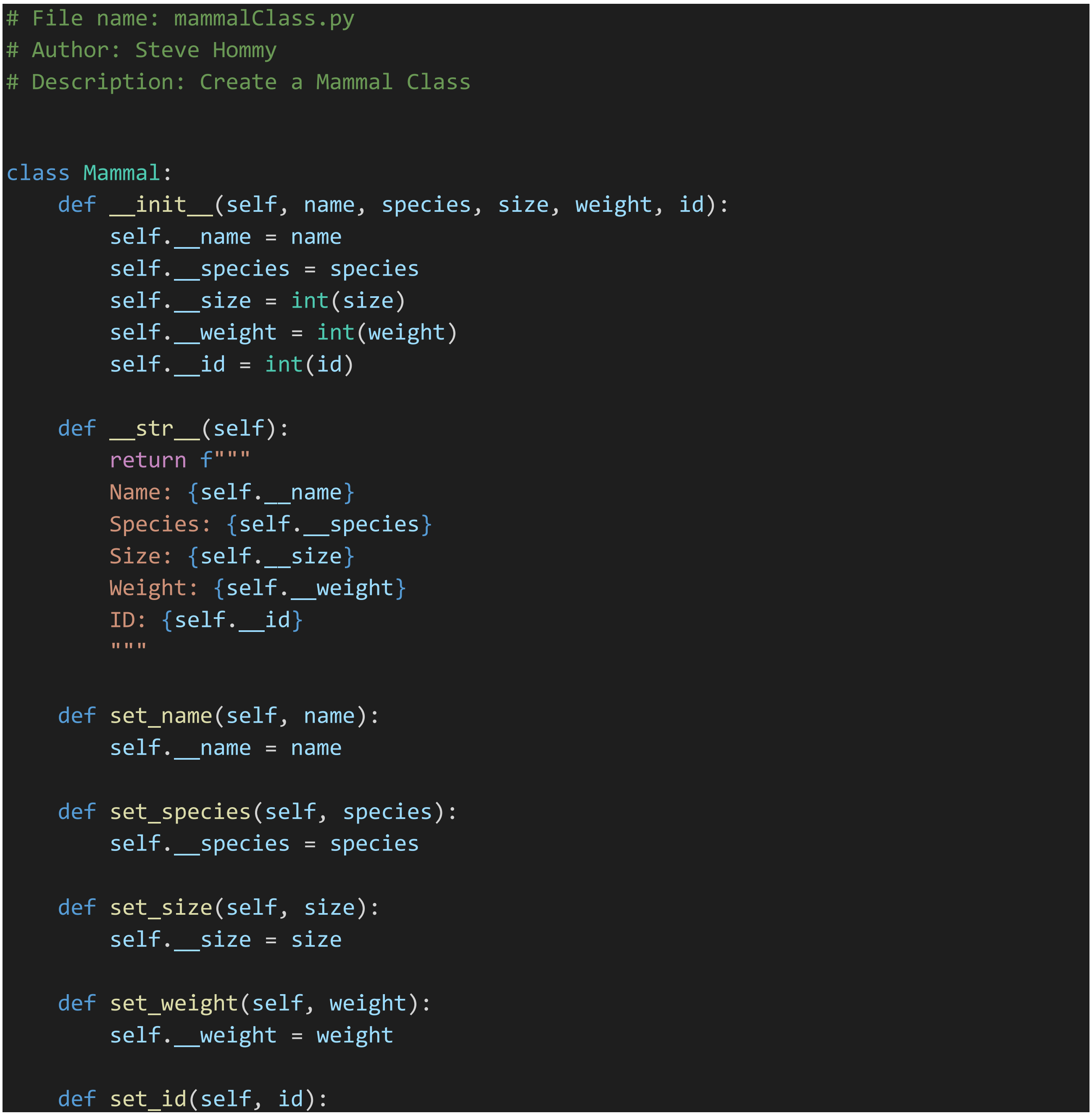
Code



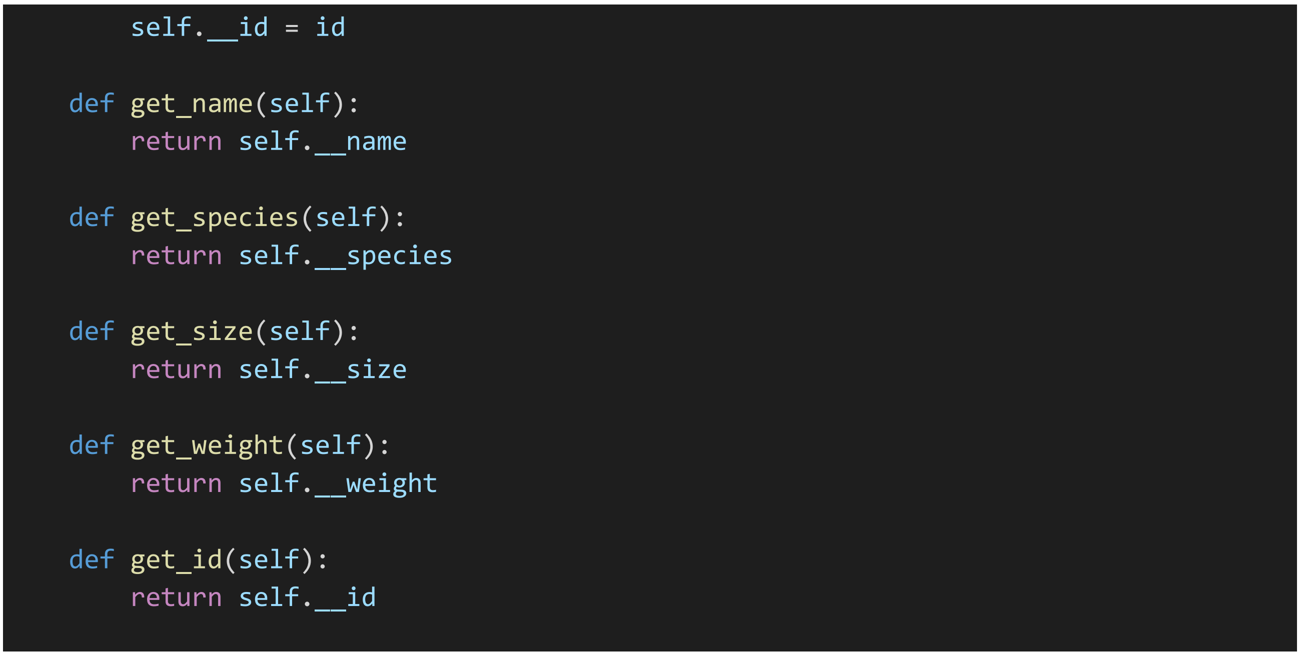
```python
# File name: mammalClass.py
# Author: Steve Hommy
# Description: Inherit Mammal Class and creating Dog Class


from mammalClass import Mammal


class Dog(Mammal):
    def __init__(self, name, species, size, weight, id, noise, diet, owner):
        Mammal.__init__(self, name, species, size, weight, id)
        self.noise = noise
        self.diet = diet
        self.owner = owner

    def __str__(self):
        return super().__str__() + f"""Noise of the animal: {self.noise}
        Diet: {self.diet}
        Owner: {self.owner}
        """

    def set_noise(self, noise):
        self.noise = noise

    def set_diet(self, diet):
        self.diet = diet
```



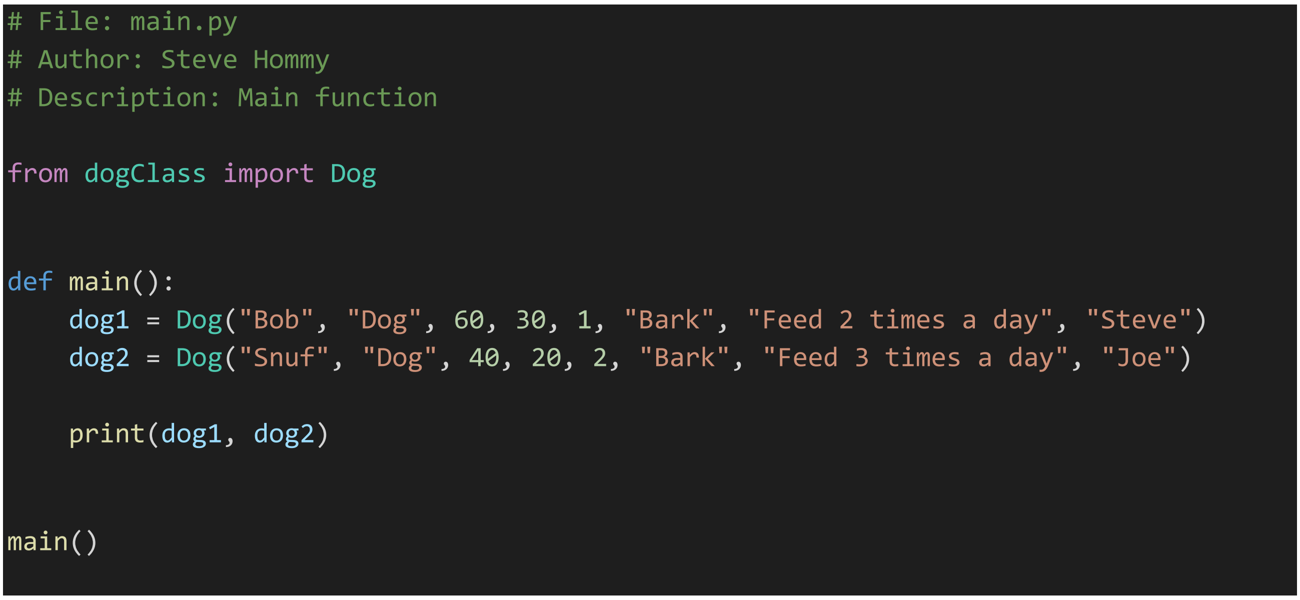
```python
    def set_owner(self, owner):
        self.owner = owner

    def get_noise(self):
        return self.noise

    def get_diet(self):
        return self.diet

    def get_owner(self):
        return self.owner
```



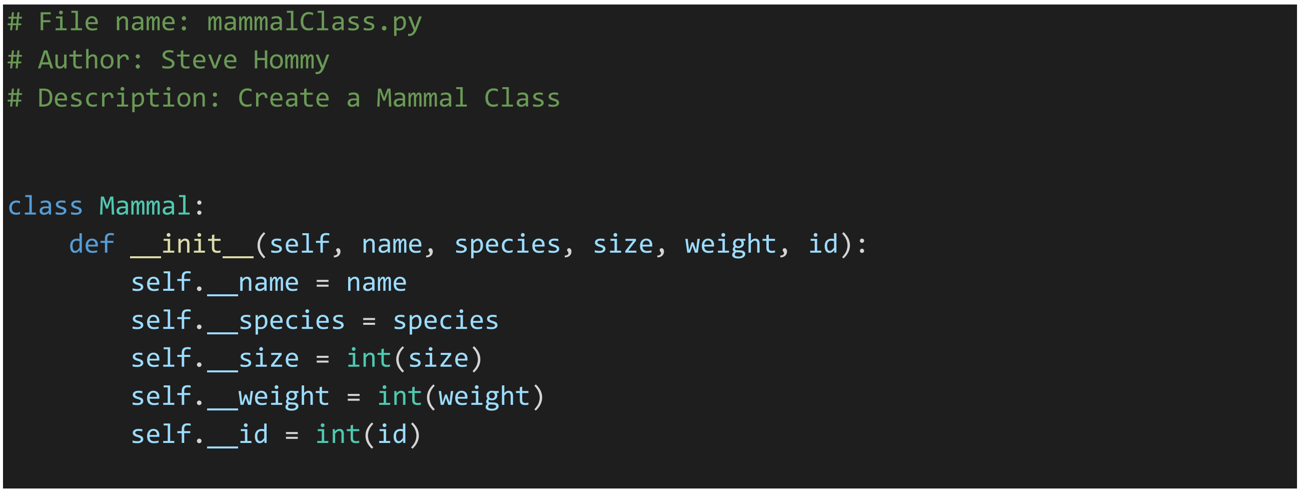
```python
# File name: mammalClass.py
# Author: Steve Hommy
# Description: Create a Mammal Class


class Mammal:
    def __init__(self, name, species, size, weight, id):
        self.__name = name
        self.__species = species
        self.__size = int(size)
        self.__weight = int(weight)
        self.__id = int(id)

    def __str__(self):
        return f"""
        Name: {self.__name}
        Species: {self.__species}
        Size: {self.__size}
        Weight: {self.__weight}
        ID: {self.__id}
        """

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

    def set_species(self, species):
        self.__species = species

    def set_size(self, size):
        self.__size = size

    def set_weight(self, weight):
        self.__weight = weight

    def set_id(self, id):
```



```
        self.__id = id

    def get_name(self):
        return self.__name

    def get_species(self):
        return self.__species

    def get_size(self):
        return self.__size

    def get_weight(self):
        return self.__weight

    def get_id(self):
        return self.__id
```

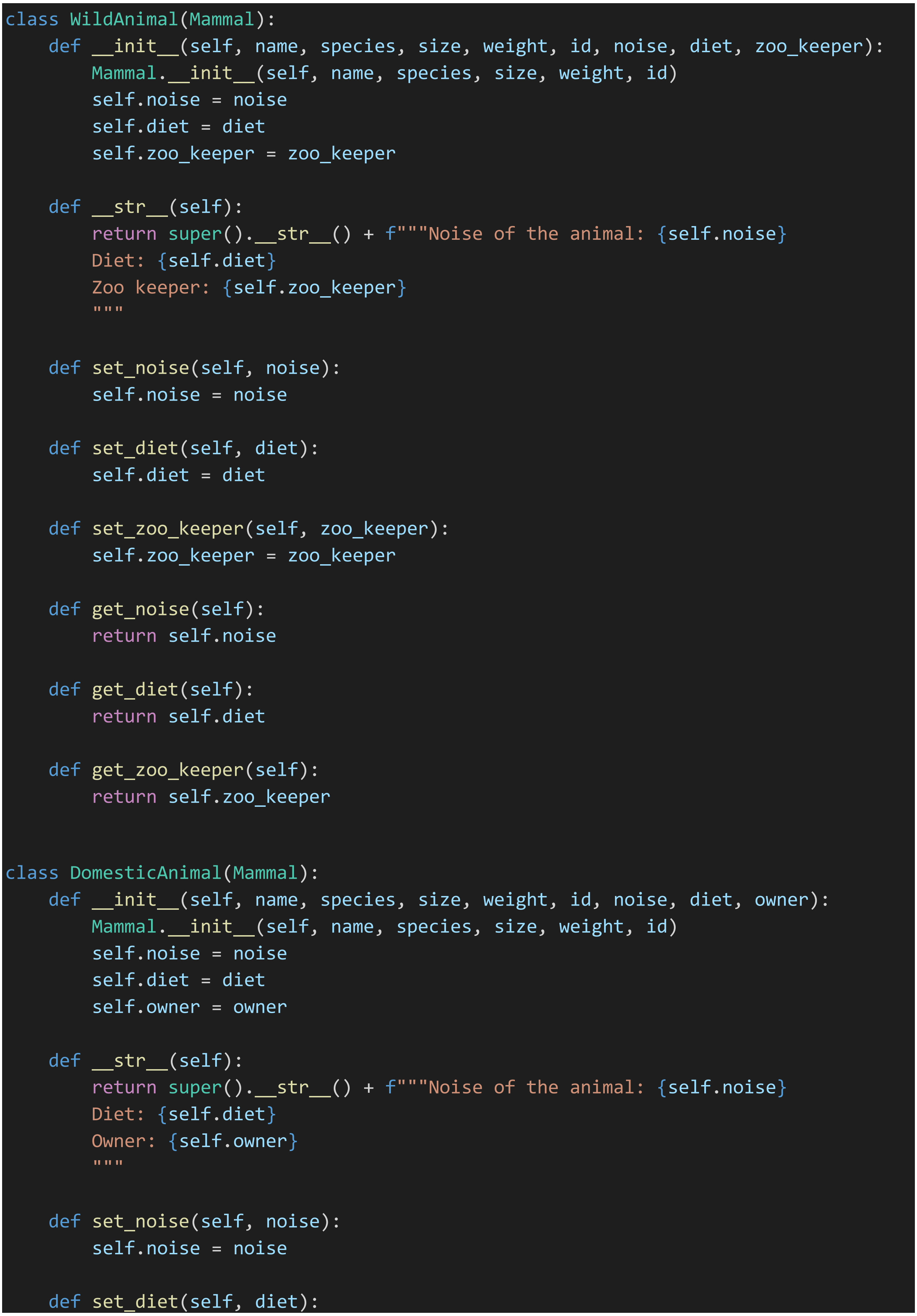


```
# File: main.py
# Author: Steve Hommy
# Description: Main function

from dogClass import Dog


def main():
    dog1 = Dog("Bob", "Dog", 60, 30, 1, "Bark", "Feed 2 times a day", "Steve")
    dog2 = Dog("Snuf", "Dog", 40, 20, 2, "Bark", "Feed 3 times a day", "Joe")

    print(dog1, dog2)


main()
```



```
# File name: mammalClass.py
# Author: Steve Hommy
# Description: Create a Mammal Class


class Mammal:
    def __init__(self, name, species, size, weight, id):
        self.__name = name
        self.__species = species
        self.__size = int(size)
        self.__weight = int(weight)
        self.__id = int(id)
    def __str__(self):
        return f"""
        Name: {self.__name}
        Species: {self.__species}
        Size: {self.__size}
        Weight: {self.__weight}
        ID: {self.__id}
        """

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

    def set_species(self, species):
        self.__species = species

    def set_size(self, size):
        self.__size = size

    def set_weight(self, weight):
        self.__weight = weight

    def set_id(self, id):
        self.__id = id

    def get_name(self):
        return self.__name

    def get_species(self):
        return self.__species

    def get_size(self):
        return self.__size

    def get_weight(self):
        return self.__weight

    def get_id(self):
        return self.__id
```

```
# File name: animalType.py
# Author: Steve Hommy
# Description: creating WildAnimal Class and DomesticAnimal Class


from mammalClass import Mammal
```



```
class WildAnimal(Mammal):
    def __init__(self, name, species, size, weight, id, noise, diet, zoo_keeper):
        Mammal.__init__(self, name, species, size, weight, id)
        self.noise = noise
        self.diet = diet
        self.zoo_keeper = zoo_keeper

    def __str__(self):
        return super().__str__() + f"""Noise of the animal: {self.noise}
        Diet: {self.diet}
        Zoo keeper: {self.zoo_keeper}
        """

    def set_noise(self, noise):
        self.noise = noise

    def set_diet(self, diet):
        self.diet = diet

    def set_zoo_keeper(self, zoo_keeper):
        self.zoo_keeper = zoo_keeper

    def get_noise(self):
        return self.noise

    def get_diet(self):
        return self.diet

    def get_zoo_keeper(self):
        return self.zoo_keeper


class DomesticAnimal(Mammal):
    def __init__(self, name, species, size, weight, id, noise, diet, owner):
        Mammal.__init__(self, name, species, size, weight, id)
        self.noise = noise
        self.diet = diet
        self.owner = owner

    def __str__(self):
        return super().__str__() + f"""Noise of the animal: {self.noise}
        Diet: {self.diet}
        Owner: {self.owner}
        """

    def set_noise(self, noise):
        self.noise = noise

    def set_diet(self, diet):
        self.diet = diet

    def set_owner(self, owner):
        self.owner = owner

    def get_noise(self):
        return self.noise

    def get_diet(self):
        return self.diet

    def get_owner(self):
        return self.owner
```

```
# File name: animals.py
# Author: Steve Hommy
# Description: creating different animal Classes


from animalType import WildAnimal, DomesticAnimal


class Dog(DomesticAnimal):
    def __init__(self, name, species, size, weight, id, noise, diet, owner, color):
        DomesticAnimal.__init__(self, name, species, size, weight, id, noise, diet, 
owner)
        self.color = color

    def __str__(self):
        return super().__str__() + f"""Color: {self.color}
        """

    def set_color(self, color):
        self.color = color

    def get_color(self):
        return self.color


class Cat(DomesticAnimal):
    def __init__(self, name, species, size, weight, id, noise, diet, owner, color):
        DomesticAnimal.__init__(self, name, species, size, weight, id, noise, diet, 
owner)
        self.color = color

    def __str__(self):
        return super().__str__() + f"""Color: {self.color}
        """
    def set_color(self, color):
        self.color = color

    def get_color(self):
        return self.color


class Tiger(WildAnimal):
    def __init__(self, name, species, size, weight, id, noise, diet, zoo_keeper,
speed):
        WildAnimal.__init__(self, name, species, size, weight, id, noise, diet,
zoo_keeper)
        self.speed = int(speed)

    def __str__(self):
        return super().__str__() + f"""Speed: {self.speed}
        """

    def set_speed(self, speed):
        self.speed = speed

    def get_speed(self):
        return self.speed


class Lion(WildAnimal):
    def __init__(self, name, species, size, weight, id, noise, diet, zoo_keeper,
speed):
        WildAnimal.__init__(self, name, species, size, weight, id, noise, diet,
zoo_keeper)
        self.speed = int(speed)

    def __str__(self):
        return super().__str__() + f"""Speed: {self.speed}
        """

    def set_speed(self, speed):
        self.speed = speed

    def get_speed(self):
        return self.speed
```

```
# File: main.py
# Author: Steve Hommy
# Description: Main function

from animals import *
```

```python
def main():
    dog = Dog("Bob", "Dog", 60, 30, 1, "Bark", "Feed 2 times a day", "Steve",
"black")
    cat = Cat("Snuf", "Cat", 20, 10, 2, "Miaw", "Feed 1 times a day", "Steve",
"brown")
    tiger = Tiger("Nie", "Tiger", 40, 60, 3, "Rawr", "Feed 3 times a day", "Joe",
50)
    lion = Lion("Max", "Lion", 50, 80, 4, "Grawr", "Feed 4 times a day", "Joe", 40)

    print(dog, cat, lion, tiger)


main()
```

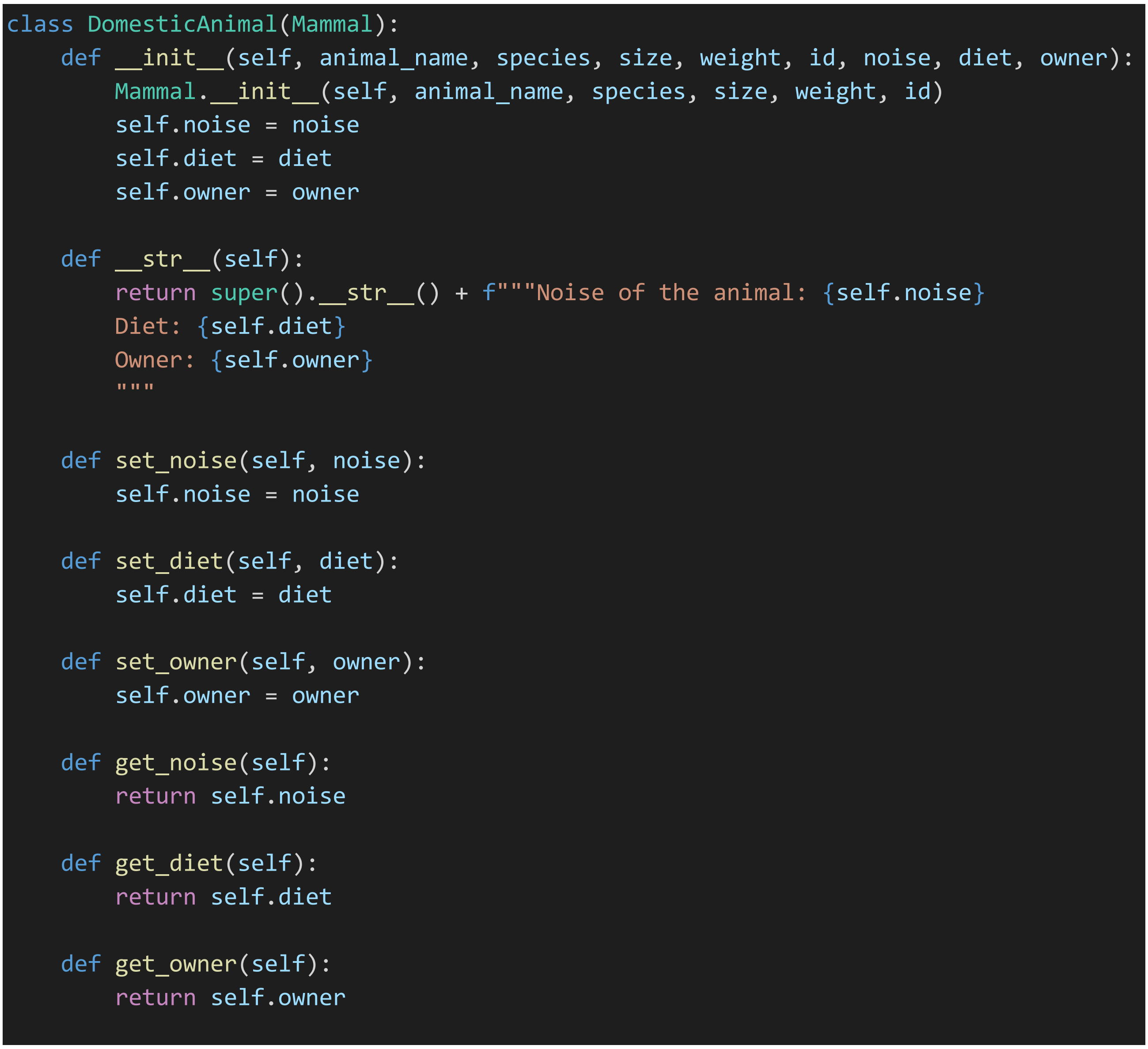
```python
# File name: participantOfOopCourse.py
# Author: Steve Hommy
# Description: Create a ParticipantOfOopCourse Class


class ParticipantOfOopCourse:
    def __init__(self, name, gender, class_number):
        self.__name = name
        self.__gender = gender
        self.__class_number = int(class_number)

    def __str__(self):
        return f"""
        Name: {self.__name}
        Gender: {self.__gender}
        Class Number: {self.__class_number}
        """

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

    def set_gender(self, gender):
        self.__gender = gender

    def set_class_number(self, class_number):
        self.__class_number = class_number

    def get_name(self):
        return self.__name

    def get_gender(self):
        return self.__gender
    def get_class_number(self):
        return self.__class_number
```

```python
# File name: studentClass.py
# Author: Steve Hommy
# Description: Inherit ParticipantOfOopCourse Class and create Student Class


from participantOfOopCourse import ParticipantOfOopCourse


class Student(ParticipantOfOopCourse):
    def __init__(self, name, gender, class_number, student_number, year_of_study):
        ParticipantOfOopCourse.__init__(self, name, gender, class_number)
        self.__student_number = int(student_number)
        self.__year_of_study = int(year_of_study)

    def __str__(self):
        return super().__str__() + f"""Student number: {self.__student_number}
        Year of study: {self.__year_of_study}
        """

    def set_student_number(self, student_number):
        self.__student_number = student_number

    def set_year_of_study(self, year_of_study):
        self.__year_of_study = year_of_study

    def get_student_number(self):
        return self.__student_number

    def get_year_of_study(self):
        return self.__year_of_study
```

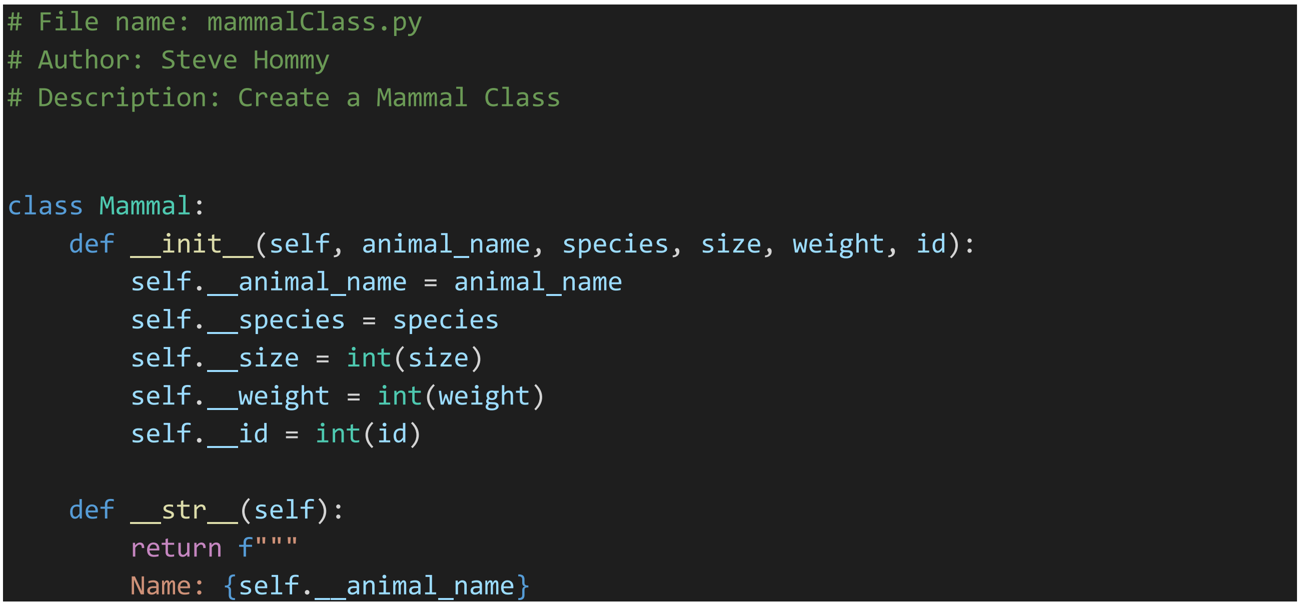
```python
# File name: teacherClass.py
# Author: Steve Hommy
# Description: Inherit ParticipantOfOopCourse Class and create Teacher Class


from participantOfOopCourse import ParticipantOfOopCourse


class Teacher(ParticipantOfOopCourse):
    def __init__(self, name, gender, class_number, academic_rank, phone_number):
        ParticipantOfOopCourse.__init__(self, name, gender, class_number)
        self.__academic_rank = academic_rank
        self.__phone_number = int(phone_number)

    def __str__(self):
        return super().__str__() + f"""Academic rank: {self.__academic_rank}
        Phone number: {self.__phone_number}
        """

    def set_academic_rank(self, academic_rank):
        self.__academic_rank = academic_rank

    def set_phone_number(self, phone_number):
        self.__phone_number = phone_number

    def get_academic_rank(self):
        return self.__academic_rank

    def get_phone_number(self):
        return self.__phone_number
```

```
# File: main.py
# Author: Steve Hommy
# Description: Main function


from studentClass import Student
from teacherClass import Teacher


def main():
    student = Student("Steve", "Male", 3, 324198, 3)
    teacher = Teacher("Sanna", "Female", 3, "Lecturer", +358954381273)

    print(student, teacher)


main()
```

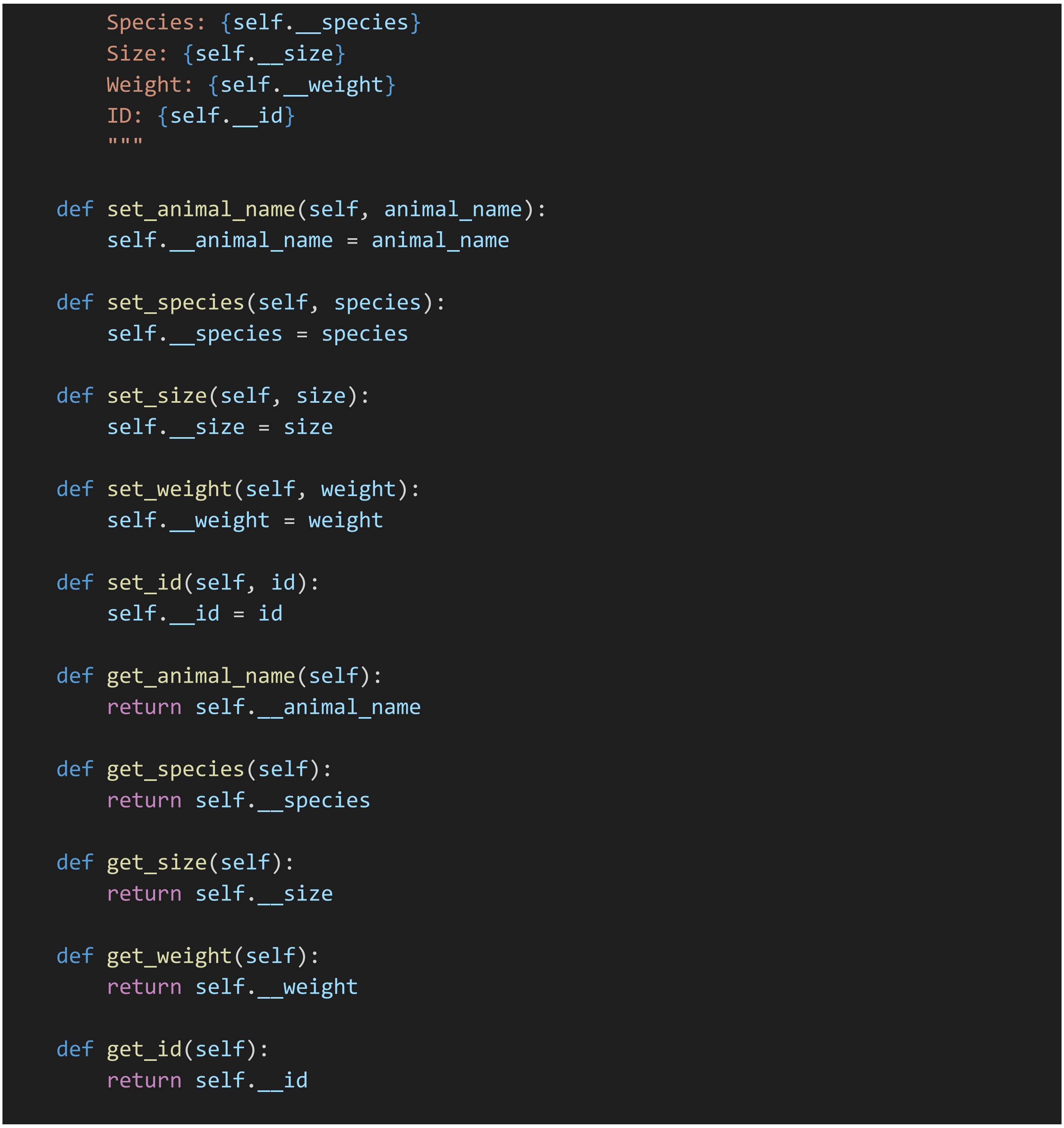
```
# File name: domesticAnimalClass.py
# Author: Steve Hommy
# Description: Inherit Mammal Class and creating DomesticAnimal Class


from mammalClass import Mammal
```



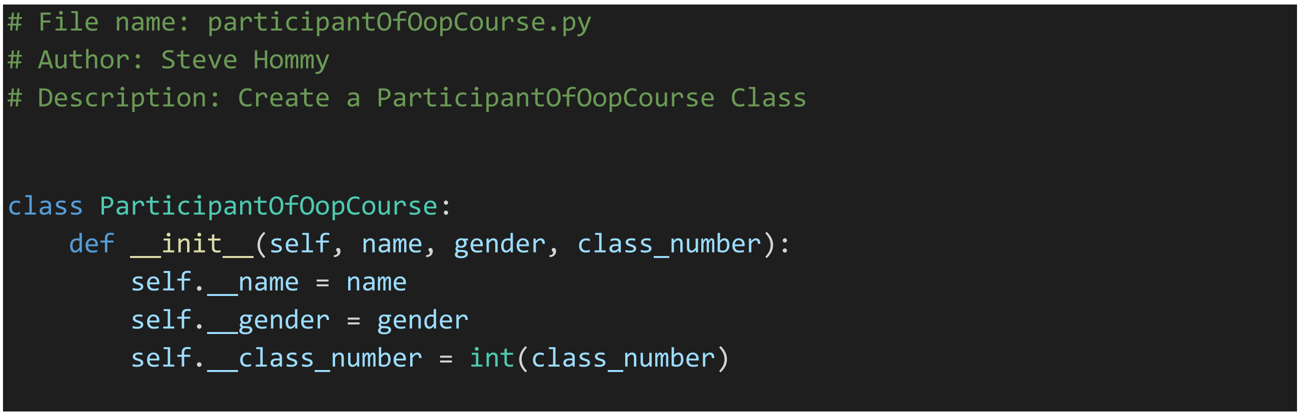
```python
class DomesticAnimal(Mammal):
    def __init__(self, animal_name, species, size, weight, id, noise, diet, owner):
        Mammal.__init__(self, animal_name, species, size, weight, id)
        self.noise = noise
        self.diet = diet
        self.owner = owner

    def __str__(self):
        return super().__str__() + f"""Noise of the animal: {self.noise}
        Diet: {self.diet}
        Owner: {self.owner}
        """

    def set_noise(self, noise):
        self.noise = noise

    def set_diet(self, diet):
        self.diet = diet

    def set_owner(self, owner):
        self.owner = owner

    def get_noise(self):
        return self.noise

    def get_diet(self):
        return self.diet

    def get_owner(self):
        return self.owner
```



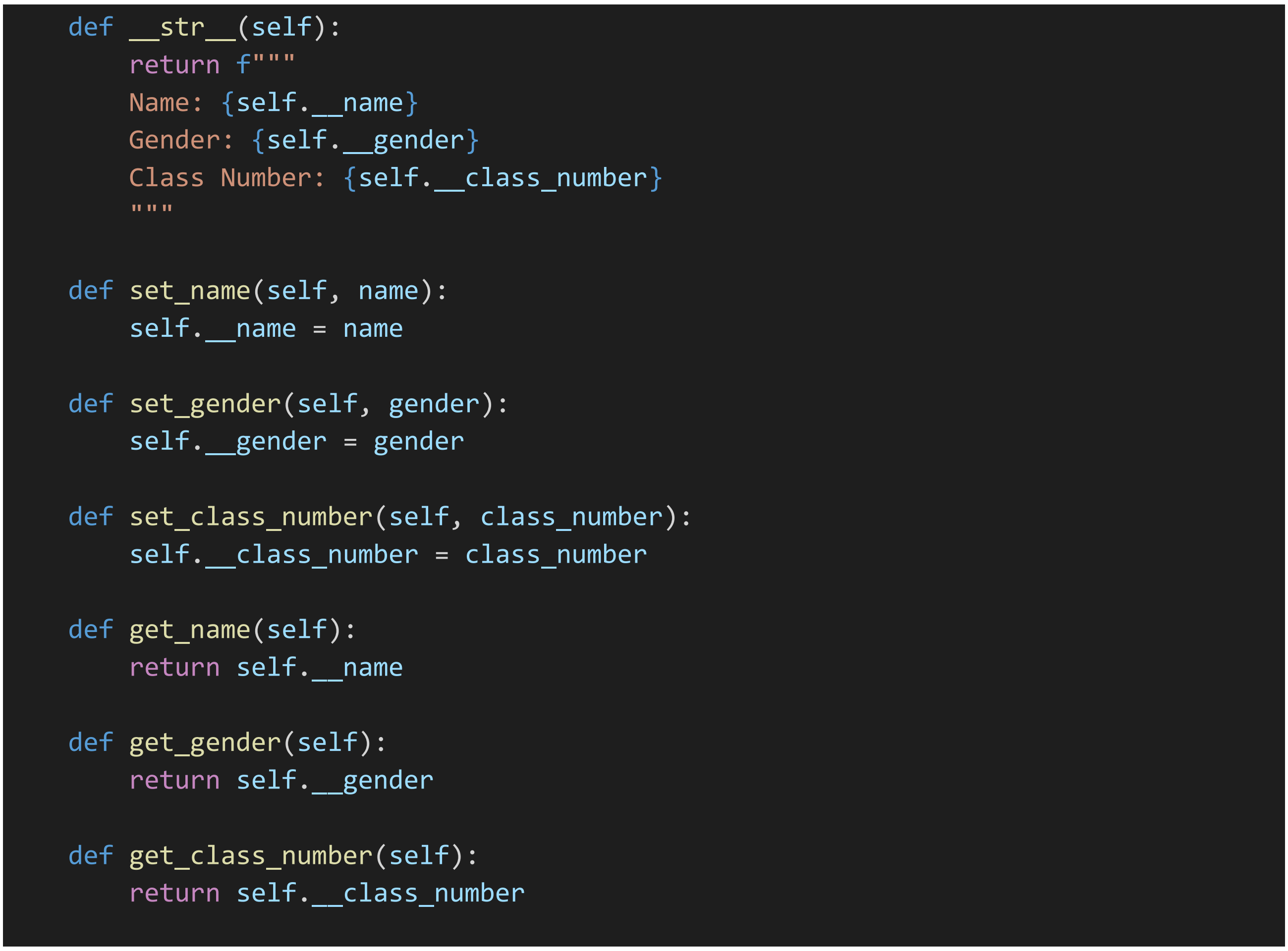
```python
# File name: mammalClass.py
# Author: Steve Hommy
# Description: Create a Mammal Class


class Mammal:
    def __init__(self, animal_name, species, size, weight, id):
        self.__animal_name = animal_name
        self.__species = species
        self.__size = int(size)
        self.__weight = int(weight)
        self.__id = int(id)

    def __str__(self):
        return f"""
        Name: {self.__animal_name}
```



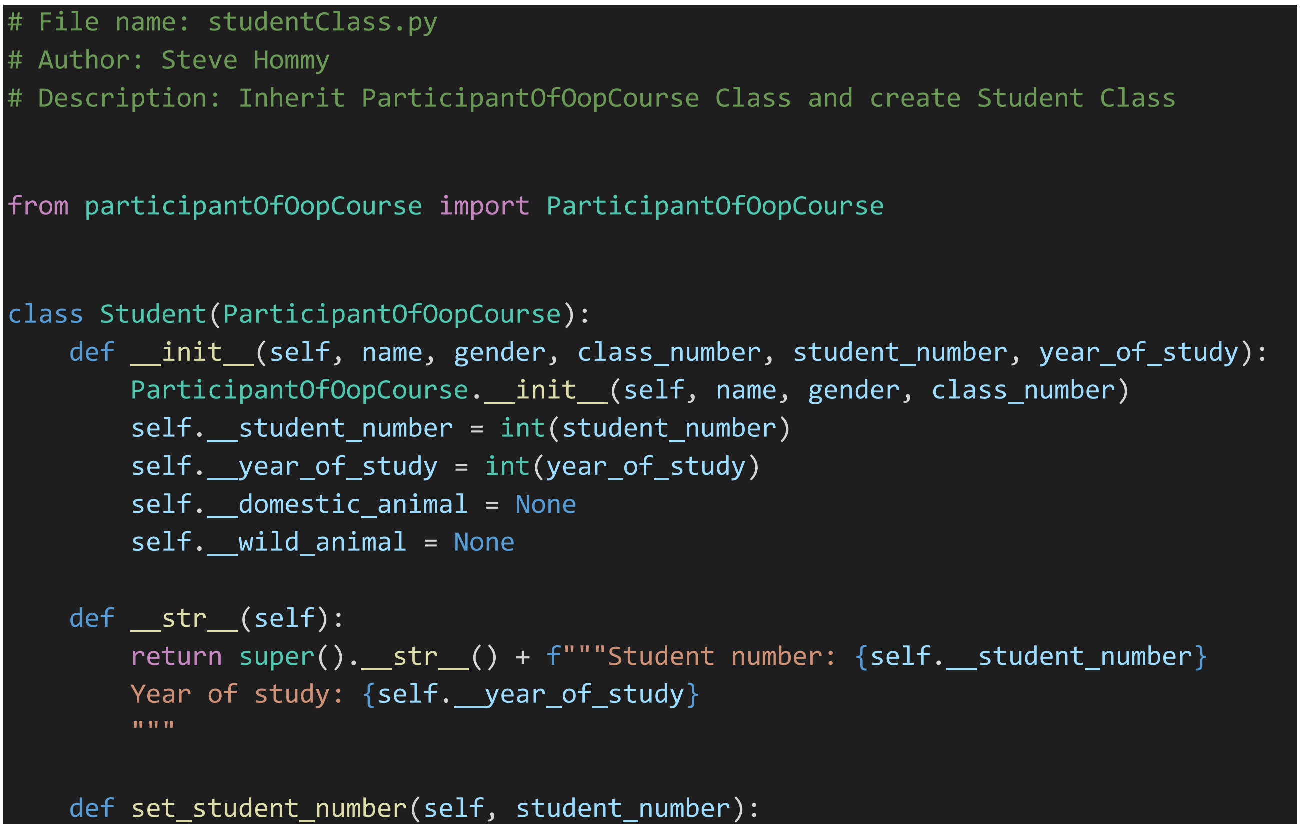
```
        Species: {self.__species}
        Size: {self.__size}
        Weight: {self.__weight}
        ID: {self.__id}
        """

    def set_animal_name(self, animal_name):
        self.__animal_name = animal_name

    def set_species(self, species):
        self.__species = species

    def set_size(self, size):
        self.__size = size

    def set_weight(self, weight):
        self.__weight = weight

    def set_id(self, id):
        self.__id = id

    def get_animal_name(self):
        return self.__animal_name

    def get_species(self):
        return self.__species

    def get_size(self):
        return self.__size

    def get_weight(self):
        return self.__weight

    def get_id(self):
        return self.__id
```



```
# File name: participantOfOopCourse.py
# Author: Steve Hommy
# Description: Create a ParticipantOfOopCourse Class


class ParticipantOfOopCourse:
    def __init__(self, name, gender, class_number):
        self.__name = name
        self.__gender = gender
        self.__class_number = int(class_number)
```



```python
    def __str__(self):
        return f"""
        Name: {self.__name}
        Gender: {self.__gender}
        Class Number: {self.__class_number}
        """

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

    def set_gender(self, gender):
        self.__gender = gender

    def set_class_number(self, class_number):
        self.__class_number = class_number

    def get_name(self):
        return self.__name

    def get_gender(self):
        return self.__gender

    def get_class_number(self):
        return self.__class_number
```



```python
# File name: studentClass.py
# Author: Steve Hommy
# Description: Inherit ParticipantOfOopCourse Class and create Student Class


from participantOfOopCourse import ParticipantOfOopCourse


class Student(ParticipantOfOopCourse):
    def __init__(self, name, gender, class_number, student_number, year_of_study):
        ParticipantOfOopCourse.__init__(self, name, gender, class_number)
        self.__student_number = int(student_number)
        self.__year_of_study = int(year_of_study)
        self.__domestic_animal = None
        self.__wild_animal = None

    def __str__(self):
        return super().__str__() + f"""Student number: {self.__student_number}
        Year of study: {self.__year_of_study}
        """

    def set_student_number(self, student_number):
        self.__student_number = student_number

    def set_year_of_study(self, year_of_study):
        self.__year_of_study = year_of_study

    def set_domestic_animal(self, domestic_animal):
        self.__domestic_animal = domestic_animal

    def set_wild_animal(self, wild_animal):
        self.__wild_animal = wild_animal

    def get_student_number(self):
        return self.__student_number

    def get_year_of_study(self):
        return self.__year_of_study

    def get_domestic_animal(self):
        return self.__domestic_animal

    def get_wild_animal(self):
        return self.__wild_animal
```

```
# File name: teacherClass.py
# Author: Steve Hommy
# Description: Inherit ParticipantOfOopCourse Class and create Teacher Class


from participantOfOopCourse import ParticipantOfOopCourse


class Teacher(ParticipantOfOopCourse):
    def __init__(self, name, gender, class_number, academic_rank, phone_number):
        ParticipantOfOopCourse.__init__(self, name, gender, class_number)
        self.__academic_rank = academic_rank
        self.__phone_number = int(phone_number)
        self.__domestic_animal = None
        self.__wild_animal = None

    def __str__(self):
        return super().__str__() + f"""Academic rank: {self.__academic_rank}
        Phone number: {self.__phone_number}
        """

    def set_academic_rank(self, academic_rank):
        self.__academic_rank = academic_rank
    def set_phone_number(self, phone_number):
        self.__phone_number = phone_number

    def set_domestic_animal(self, domestic_animal):
        self.__domestic_animal = domestic_animal

    def set_wild_animal(self, wild_animal):
        self.__wild_animal = wild_animal

    def get_academic_rank(self):
        return self.__academic_rank

    def get_phone_number(self):
        return self.__phone_number

    def get_domestic_animal(self):
        return self.__domestic_animal

    def get_wild_animal(self):
        return self.__wild_animal
```

```
# File name: wildAnimalClass.py
# Author: Steve Hommy
# Description: Inherit Mammal Class and creating WildAnimal Class


from mammalClass import Mammal


class WildAnimal(Mammal):
    def __init__(self, animal_name, species, size, weight, id, noise, diet,
zoo_keeper):
        Mammal.__init__(self, animal_name, species, size, weight, id)
        self.noise = noise
        self.diet = diet
        self.zoo_keeper = zoo_keeper

    def __str__(self):
        return super().__str__() + f"""Noise of the animal: {self.noise}
        Diet: {self.diet}
        Zoo keeper: {self.zoo_keeper}
        """

    def set_noise(self, noise):
        self.noise = noise

    def set_diet(self, diet):
        self.diet = diet

    def set_zoo_keeper(self, zoo_keeper):
        self.zoo_keeper = zoo_keeper

    def get_noise(self):
        return self.noise

    def get_diet(self):
        return self.diet

    def get_zoo_keeper(self):
        return self.zoo_keeper
```

```
# File: main.py
# Author: Steve Hommy
# Description: Main function


from teacherClass import Teacher
from studentClass import Student
from domesticAnimalClass import DomesticAnimal
from wildAnimalClass import WildAnimal


def main():
    student = Student("Steve", "Male", 3, 324198, 3)
    teacher = Teacher("Sanna", "Female", 3, "Lecturer", +358954381273)
    dog = DomesticAnimal("Bob", "Dog", 60, 30, 1, "Bark", "Feed 2 times a day",
"Steve")
    cat = DomesticAnimal("Snuf", "Cat", 20, 10, 2, "Miaw", "Feed 1 times a day",
"Sanna")
    tiger = WildAnimal("Nie", "Tiger", 40, 60, 3, "Rawr", "Feed 3 times a day",
"Joe")
    lion = WildAnimal("Max", "Lion", 50, 80, 4, "Grawr", "Feed 4 times a day",
"Joe")

    student.set_domestic_animal(dog)
    student.set_wild_animal(tiger)

    teacher.set_domestic_animal(cat)
    teacher.set_wild_animal(lion)

    print(f"""{student}
    domestic animal is:
    {student.get_domestic_animal()}
    and wild animal:
```

```
    {student.get_wild_animal()}""")

    print(f"""{teacher}
    domestic animal is:
    {teacher.get_domestic_animal()}
    and wild animal:
    {teacher.get_wild_animal()}""")


main()
```

```
# File name: domesticAnimalClass.py
# Author: Steve Hommy
# Description: Inherit Mammal Class and creating DomesticAnimal Class


from mammalClass import Mammal


class DomesticAnimal(Mammal):
    def __init__(self, animal_name, species, size, weight, id, noise, diet, owner):
        Mammal.__init__(self, animal_name, species, size, weight, id)
        self.noise = noise
        self.diet = diet
        self.owner = owner

    def __str__(self):
        return super().__str__() + f"""Noise of the animal: {self.noise}
        Diet: {self.diet}
        Owner: {self.owner}
        """

    def set_noise(self, noise):
        self.noise = noise

    def set_diet(self, diet):
        self.diet = diet

    def set_owner(self, owner):
        self.owner = owner

    def get_noise(self):
        return self.noise

    def get_diet(self):
        return self.diet

    def get_owner(self):
        return self.owner
```

```
# File name: mammalClass.py
# Author: Steve Hommy
# Description: Create a Mammal Class


class Mammal:
    def __init__(self, animal_name, species, size, weight, id):
        self.__animal_name = animal_name
        self.__species = species
        self.__size = int(size)
        self.__weight = int(weight)
        self.__id = int(id)

    def __str__(self):
        return f"""
        Name: {self.__animal_name}
        Species: {self.__species}
        Size: {self.__size}
        Weight: {self.__weight}
        ID: {self.__id}
        """

    def set_animal_name(self, animal_name):
        self.__animal_name = animal_name

    def set_species(self, species):
        self.__species = species

    def set_size(self, size):
        self.__size = size

    def set_weight(self, weight):
        self.__weight = weight

    def set_id(self, id):
        self.__id = id

    def get_animal_name(self):
        return self.__animal_name

    def get_species(self):
        return self.__species

    def get_size(self):
        return self.__size
```

```python

    def get_weight(self):
        return self.__weight

    def get_id(self):
        return self.__id
```

```python
# File name: participantOfOopCourse.py
# Author: Steve Hommy
# Description: Create a ParticipantOfOopCourse Class


class ParticipantOfOopCourse:
    def __init__(self, name, gender, class_number):
        self.__name = name
        self.__gender = gender
        self.__class_number = int(class_number)

    def __str__(self):
        return f"""
        Name: {self.__name}
        Gender: {self.__gender}
        Class Number: {self.__class_number}
        """

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

    def set_gender(self, gender):
        self.__gender = gender

    def set_class_number(self, class_number):
        self.__class_number = class_number

    def get_name(self):
        return self.__name

    def get_gender(self):
        return self.__gender

    def get_class_number(self):
        return self.__class_number
```

```python
# File name: studentClass.py
# Author: Steve Hommy
# Description: Inherit ParticipantOfOopCourse Class and create Student Class
```

```
from participantOfOopCourse import ParticipantOfOopCourse


class Student(ParticipantOfOopCourse):
    def __init__(self, name, gender, class_number, student_number, year_of_study):
        ParticipantOfOopCourse.__init__(self, name, gender, class_number)
        self.__student_number = int(student_number)
        self.__year_of_study = int(year_of_study)
        self.__domestic_animal = None
        self.__wild_animal = None

    def __str__(self):
        return super().__str__() + f"""Student number: {self.__student_number}
        Year of study: {self.__year_of_study}
        """

    def set_student_number(self, student_number):
        self.__student_number = student_number

    def set_year_of_study(self, year_of_study):
        self.__year_of_study = year_of_study

    def set_domestic_animal(self, domestic_animal):
        self.__domestic_animal = domestic_animal

    def set_wild_animal(self, wild_animal):
        self.__wild_animal = wild_animal

    def get_student_number(self):
        return self.__student_number

    def get_year_of_study(self):
        return self.__year_of_study

    def get_domestic_animal(self):
        return self.__domestic_animal

    def get_wild_animal(self):
        return self.__wild_animal
```

```
# File name: teacherClass.py
# Author: Steve Hommy
# Description: Inherit ParticipantOfOopCourse Class and create Teacher Class


from participantOfOopCourse import ParticipantOfOopCourse
```

```
class Teacher(ParticipantOfOopCourse):
    def __init__(self, name, gender, class_number, academic_rank, phone_number):
        ParticipantOfOopCourse.__init__(self, name, gender, class_number)
        self.__academic_rank = academic_rank
        self.__phone_number = int(phone_number)
        self.__domestic_animal = None
        self.__wild_animal = None

    def __str__(self):
        return super().__str__() + f"""Academic rank: {self.__academic_rank}
        Phone number: {self.__phone_number}
        """

    def set_academic_rank(self, academic_rank):
        self.__academic_rank = academic_rank

    def set_phone_number(self, phone_number):
        self.__phone_number = phone_number

    def set_domestic_animal(self, domestic_animal):
        self.__domestic_animal = domestic_animal

    def set_wild_animal(self, wild_animal):
        self.__wild_animal = wild_animal

    def get_academic_rank(self):
        return self.__academic_rank

    def get_phone_number(self):
        return self.__phone_number

    def get_domestic_animal(self):
        return self.__domestic_animal

    def get_wild_animal(self):
        return self.__wild_animal
```

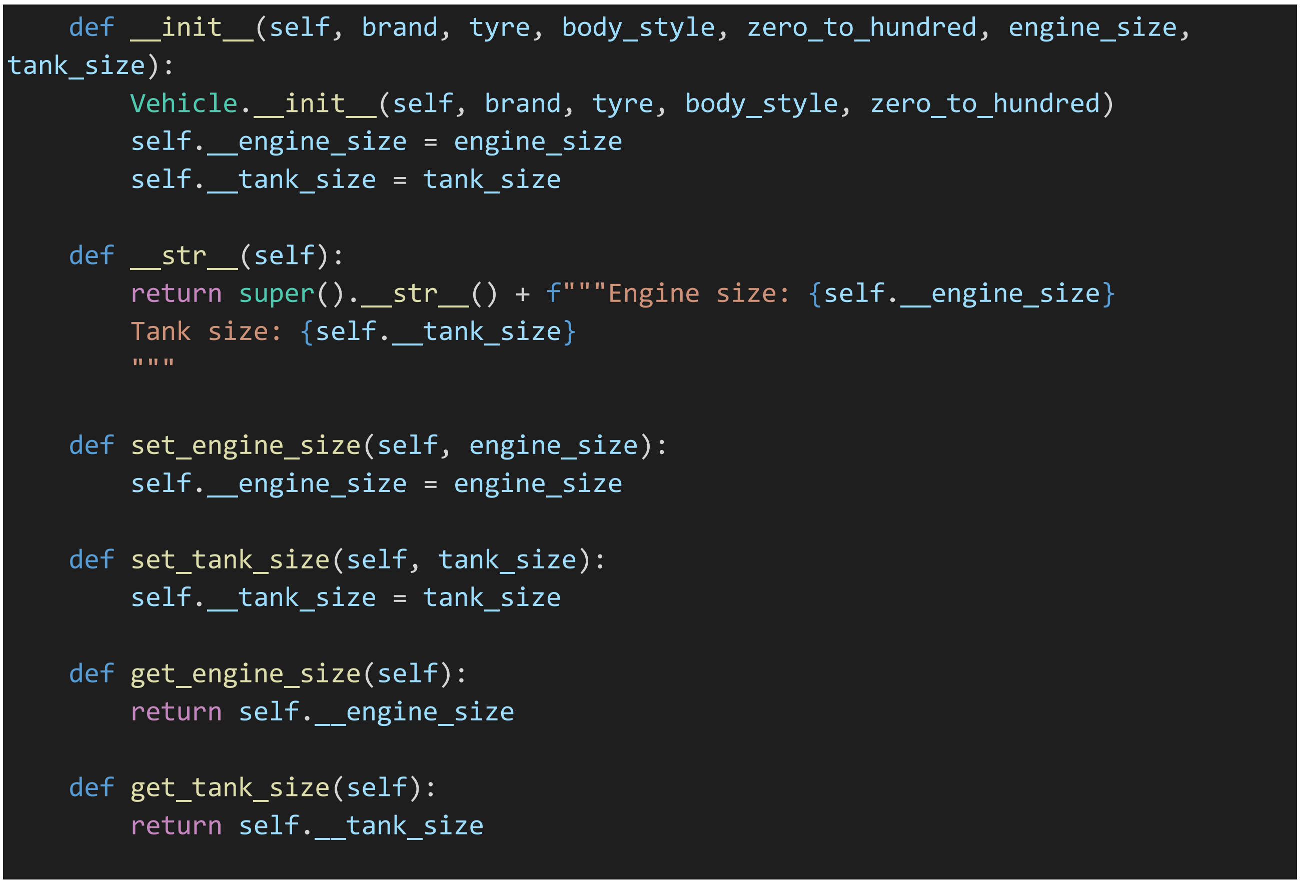
```
# File name: wildAnimalClass.py
# Author: Steve Hommy
# Description: Inherit Mammal Class and creating WildAnimal Class


from mammalClass import Mammal


class WildAnimal(Mammal):
    def __init__(self, animal_name, species, size, weight, id, noise, diet,
zoo_keeper):
        Mammal.__init__(self, animal_name, species, size, weight, id)
        self.noise = noise
        self.diet = diet
        self.zoo_keeper = zoo_keeper

    def __str__(self):
        return super().__str__() + f"""Noise of the animal: {self.noise}
        Diet: {self.diet}
        Zoo keeper: {self.zoo_keeper}
        """

    def set_noise(self, noise):
        self.noise = noise

    def set_diet(self, diet):
        self.diet = diet

    def set_zoo_keeper(self, zoo_keeper):
        self.zoo_keeper = zoo_keeper

    def get_noise(self):
        return self.noise

    def get_diet(self):
        return self.diet

    def get_zoo_keeper(self):
        return self.zoo_keeper
```

```
# File: main.py
# Author: Steve Hommy
# Description: Main function


from teacherClass import Teacher
from studentClass import Student
from domesticAnimalClass import DomesticAnimal
from wildAnimalClass import WildAnimal


def main():
    student = Student("Steve", "Male", 3, 324198, 3)
    teacher = Teacher("Sanna", "Female", 3, "Lecturer", +358954381273)
    dog = DomesticAnimal("Bob", "Dog", 60, 30, 1, "Bark", "Feed 2 times a day",
"Steve")
```

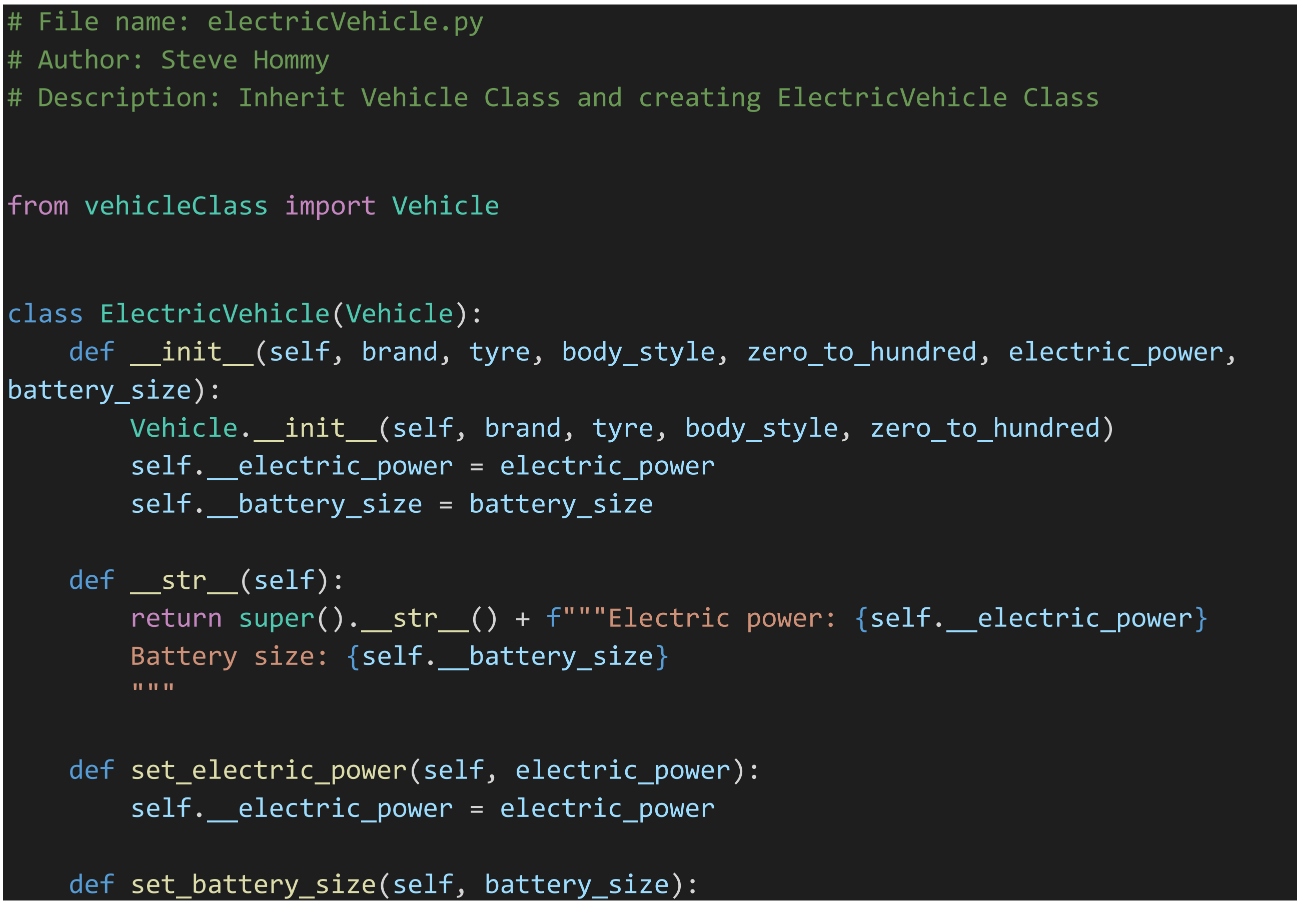
```
    hamster = DomesticAnimal("Pier", "Hamster", 2, 5, 3, "Squeek", "Feed 1 times a
day", "Steve")
    cat = DomesticAnimal("Snuf", "Cat", 20, 10, 2, "Miaw", "Feed 1 times a day",
"Sanna")
    rabbit = DomesticAnimal("Nown", "Rabbit", 5, 10, 4, "Shhh", "Feed 1.5 times a
day", "Sanna")
    tiger = WildAnimal("Nie", "Tiger", 40, 60, 3, "Rawr", "Feed 3 times a day",
"Joe")
    lion = WildAnimal("Max", "Lion", 50, 80, 4, "Grawr", "Feed 4 times a day",
"Joe")

    student_domestic_animals = [dog, hamster]
    student_wild_animals = [tiger, lion]
    teacher_domestic_animals = [cat, rabbit]
    teacher_wild_animals = [lion, tiger]

    student.set_domestic_animal(student_domestic_animals)
    student.set_wild_animal(student_wild_animals)

    teacher.set_domestic_animal(teacher_domestic_animals)
    teacher.set_wild_animal(teacher_wild_animals)

    print(f"""{student}
    domestic animals are:""")
    for student_domestic in student.get_domestic_animal():
        print(student_domestic)
    print("and wild animals are:")
    for student_wild in student.get_wild_animal():
        print(student_wild)

    print(f"""{teacher}
    domestic animals are:""")
    for teacher_domestic in teacher.get_domestic_animal():
        print(teacher_domestic)
    print("and wild animals are:")
    for teacher_wild in teacher.get_wild_animal():
        print(teacher_wild)


main()
```

```
# File name: vehicleClass.py
# Author: Steve Hommy
# Description: Create a Vehicle Class


class Vehicle:
    def __init__(self, brand, tyre, body_style, zero_to_hundred):
        self.__brand = brand
        self.__tyre = tyre
        self.__body_style = body_style
        self.__zero_to_hundred = float(zero_to_hundred)

    def __str__(self):
        return f"""
        Brand: {self.__brand}
        Tyre: {self.__tyre}
        Body style: {self.__body_style}
        0 to 100 in: {self.__zero_to_hundred} seconds
        """

    def set_brand(self, brand):
        self.__brand = brand

    def set_tyre(self, tyre):
        self.__tyre = tyre

    def set_body_style(self, body_style):
        self.__body_style = body_style

    def set_zero_to_hundred(self, zero_to_hundred):
        self.__zero_to_hundred = zero_to_hundred

    def get_brand(self):
        return self.__brand

    def get_tyre(self):
        return self.__tyre

    def get_body_style(self):
        return self.__body_style

    def get_zero_to_hundred(self):
        return self.__zero_to_hundred
```

```
# File name: petrolVehicle.py
# Author: Steve Hommy
# Description: Inherit Vehicle Class and creating PetrolVehicle Class


from vehicleClass import Vehicle


class PetrolVehicle(Vehicle):
```



```
    def __init__(self, brand, tyre, body_style, zero_to_hundred, engine_size,
tank_size):
        Vehicle.__init__(self, brand, tyre, body_style, zero_to_hundred)
        self.__engine_size = engine_size
        self.__tank_size = tank_size

    def __str__(self):
        return super().__str__() + f"""Engine size: {self.__engine_size}
        Tank size: {self.__tank_size}
        """

    def set_engine_size(self, engine_size):
        self.__engine_size = engine_size

    def set_tank_size(self, tank_size):
        self.__tank_size = tank_size

    def get_engine_size(self):
        return self.__engine_size

    def get_tank_size(self):
        return self.__tank_size
```



```
# File name: electricVehicle.py
# Author: Steve Hommy
# Description: Inherit Vehicle Class and creating ElectricVehicle Class


from vehicleClass import Vehicle


class ElectricVehicle(Vehicle):
    def __init__(self, brand, tyre, body_style, zero_to_hundred, electric_power,
battery_size):
        Vehicle.__init__(self, brand, tyre, body_style, zero_to_hundred)
        self.__electric_power = electric_power
        self.__battery_size = battery_size

    def __str__(self):
        return super().__str__() + f"""Electric power: {self.__electric_power}
        Battery size: {self.__battery_size}
        """

    def set_electric_power(self, electric_power):
        self.__electric_power = electric_power

    def set_battery_size(self, battery_size):
        self.__battery_size = battery_size

    def get_electric_power(self):
        return self.__electric_power

    def get_battery_size(self):
        return self.__battery_size
```

```
# File: main.py
# Author: Steve Hommy
# Description: Main function


from petrolVehicle import PetrolVehicle
from electricVehicle import ElectricVehicle


def main():
    honda = PetrolVehicle("Honda", "Continental", "Hatchback", 8.5, "1.6l", "100l")
    tesla = ElectricVehicle("Tesla", "Nokia", "Sedan", 4.5, "250W", "1000 000A")

    print("Our first car is:", honda)
    print("Our second car is:", tesla)

    how_fast_dict = {
        honda.get_brand(): honda.get_zero_to_hundred(),
        tesla.get_brand(): tesla.get_zero_to_hundred()
    }

    for key in how_fast_dict:
        print(key, "will reach 0 to 100 in", how_fast_dict[key], "seconds")


main()
```