**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  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 | 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 |
| **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: 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: 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 |
| **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: 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  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 | 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  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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Diagram

Description automatically generated

A picture containing text, receipt

Description automatically generated

Diagram

Description automatically generated

Diagram, box and whisker chart

Description automatically generated

Diagram, timeline

Description automatically generated

# Code

# 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

    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, 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 \*

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()

# 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

# 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

# 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

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

    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)

    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

# 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

    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)

    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

# 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()