

Assignment #5: The Barnyard

Due Date:

1. Objective

This assignment heavily tests the basics of OOP by reinforcing inheritance, classes, objects, attributes, and constructor methods. This assignment will test these concepts in an intuitive and easy to implement way.

2. Project Description

The goal of this project is to implement a farm managing program. The program will consist of allowing the user to see how many animals along with which specific animals they currently have in their farm. The user will be able to release any of their currently owned animals as well as add to their current animal list. Each of the animals will have unique attributes that are defined by the user.

3. Function & Class Restrictions

Classes must have a `__init__` function as their constructor.

4. Required Classes

- **Animal** – The Animal class will be the parent class of all other classes used in this program. The functions that the Animal class must have are described in the “Required Functions” section of this document. The Animal class will have the following attributes:
 - height – The height of the animal in feet. (Must be stored as a floating-point number)
 - weight – The weight of the animal in pounds. (Must be stored as a floating-point number)
 - age – The age of the animal in years. (Must be stored as a floating-point number)
 - name – The name of the animal. (Must be stored as a string)
- **Cow** – The Cow class is a child class of Animal (Cow inherits from Animal). The functions that the Cow class must have are described in the “Required Functions” section of this document. The Cow class will have the following attributes:
 - gallonsOfMilkADay – How many gallons of milk this particular cow produces a day. (Must be stored as a floating point number)
 - isPregnant – Whether or not this cow is pregnant. (Must be stored as a Boolean)The Cow class inherits from Animal and will also inherit the attributes in the Animal class.
- **Chicken** – The Chicken class is a child class of Animal (Chicken inherits from Animal). The functions that the Chicken class must have are described in the “Required Functions” section of this document. The Chicken class will have the following attributes:
 - eggsADay – How many eggs this particular chicken produces a day. (Must be stored as an integer)
 - isPregnant – Whether or not this chicken is pregnant. (Must be stored as a Boolean)The Chicken class inherits from Animal and will also inherit the attributes in the Animal class.

- Horse – The Horse class is a child class of Animal (Horse inherits from Animal). The functions that the Horse class must have are described in the “Required Functions” section of this document. The Horse class will have the following attributes:
 - maxSpeed – How fast this horse can run in mph. (Must be stored as a floating point number)
 - isRideable – Whether or not this horse can be ridden. (Must be stored as a Boolean)
 The Horse class inherits from Animal and will also inherit the attributes in the Animal class.
- Dog – The Dog class is a child class of Animal (Dog inherits from Animal). The functions that the Dog class must have are described in the “Required Functions” section of this document. The Dog class will have the following attributes:
 - breed – What breed the dog is. (Must be stored as a string)
 - isFriendly – Whether or not this dog is friendly. (Must be stored as a Boolean)
 The Dog class inherits from Animal and will also inherit the attributes in the Animal class.

5. Required Functions

OVERALL FUNCTIONS

- menuFunction() – Prints out a menu. The menu must follow the following format exactly:

```
Welcome to the Barnyard!

Please select an option.
1. Print Barnyard!
2. Add animal to Barnyard!
3. Remove animal from Barnyard!
4. Exit
=====
```

If option 1 is selected, then the printBarnyard function must be called with the current list of animals as its parameter. If option 2 is selected, then the addAnimal function must be called with the list of animals as its parameter. If option 3 is selected, then the remove animal function is called with the list of animals as its parameter. If option 4 is selected, then the program stops. The program should loop continuously until option 4 is selected.

- addAnimal(listOfAnimals) – This function is called if the add animal option is selected from the menu. The function will prompt the user for the information of the new animal and then add it to the list of animals given (listOfAnimals). When the function finishes executing it should add the newly created instance of the specified animal to the list of animals. The function should first print the following menu:

```
You have selected to add an animal to your Barnyard!
Please select which kind of animal you would like to add
1. Cow
2. Chicken
3. Horse
4. Dog
=====
```

Afterwards, the user will be prompted for the information of the animal. The information that will be asked of the user will depend on the type of animal selected. The user must be prompted for all the necessary information required to create an instance of that kind of animal. For example, the following prompts are made in response to a Cow being selected as the desired animal.

```
You have selected to add an animal to your Barnyard!
Please select which kind of animal you would like to add:
    1. Cow
    2. Chicken
    3. Horse
    4. Dog
=====
1
You have selected to add a Cow
How much does the cow weight?
375
How tall is the cow?
5.6
How old is the cow?
6.3
What is the name of the cow?
Annabelle
How many gallons of milk does this cow produce a day?
3.2
Is this cow currently pregnant? (Y/N)
Y
Congratulations, on your new cow!
You have added a cow with the following attributes to your barnyard
    height : 5.6 ft
    weight : 375 lb
    age : 6.3 years old
    name : Annabelle
    gallosnOfMilkADay : 3.2 gal
    isPregnant : True
=====
Welcome to the Barnyard!

Please select an option.
    1. Print Barnyard
    2. Add animal to Barnyard
    3. Remove animal from Barnyard
    4. Exit
=====
```

You must create your own prompts for each animal. Your prompts must gain all the required information and must match the style of the ones shown in the examples. You must then print out all of the information of the newly created animal, add the newly created animal to the animal list, and then return to the main menu.

- `removeAnimal(listOfAnimals)` – This function is called if the remove option is selected from the menu. The function will print out the name of the animal as well as its type in a numbered list. The user will then select the animal they would like to remove from the barnyard. After the user has made their selection, the animal will be removed from the barnyard by deleting it from the list of animals. The program must then go back to the main menu. The following is the menu that must be printed by the `removeAnimal` function as well as how it must operate.

```

You have selected to remove an animal to your Barnyard!
Please select which animal you would like to remove:
    1. Annabelle (Cow)
    2. Roger (Dog)
    3. Rover (Dog)
    4. Steve (Horse)
    5. Calypso (Cow)
=====
3
You have selected to remove Rover (Dog) from your Barnyard
=====
Welcome to the Barnyard!

Please select an option.
    1. Print Barnyard
    2. Add animal to Barnyard
    3. Remove animal from Barnyard
    4. Exit
=====

```

- `printBarnyard(listOfAnimals)` – This function is called if the print barnyard option is selected and will print all the current animals in the barnyard. If there are currently no animals in the barnyard then the function must print out “You have an empty barnyard.” Otherwise, the function must print out all of the animals in the barnyard with their attributes. The `isInstance()` function will be very useful here. The `printBarnyard` function must operate as shown below:

```

You have selected to print your Barnyard!
    1. Annabelle (Cow)
height : 5.6 ft
weight : 375lb
age : 6.2 years old
name : Annabelle
gallonsOfMilkADay : 3.2 gal
isPregnant : True
=====
    2. Roger (Dog)
height : 2.8 ft
weight : 80 lb
age : 3.0 years old
name : Roger
breed : Golden Retriever
isFriendly : True
=====
    3. Steve (Horse)
height : 6.1 ft
weight : 392 lb
age : 12.2 years old
name : Steve
maxSpeed : 31.3 mph
isRideable : True
=====
    4. Calypso (Cow)
height : 5.1 ft
weight : 330lb
age : 4.1 years old
name : Calypso
gallonsOfMilkADay : 3.6 gal
isPregnant : False
=====
Welcome to the Barnyard!

Please select an option.
    1. Print Barnyard
    2. Add animal to Barnyard
    3. Remove animal from Barnyard
    4. Exit
=====

```

ANIMAL FUNCTIONS

1. `def __init__(self, height, weight, age, name):` This function will be the constructor function of the animal class.
2. `def generalInfo(self):` This function will print out the info of this animal in the following format:

```
height : 2.8 ft
weight : 80 lb
age : 3.0 years old
name : Roger
```

With the information relevant to that specific animal.

COW FUNCTIONS

1. `def __init__(self, height, weight, age, name, gallonsOfMilkADay, isPregnant):` This function will be the constructor function of the cow class.
2. `def cowInfo(self):` This function will print out the info of this cow in the following format:

```
height : 5.6 ft
weight : 375lb
age : 6.2 years old
name : Annabelle
gallonsOfMilkADay : 3.2 gal
isPregnant : True
```

With the information relevant to that specific cow.

3. `def moo(self):` This function will print "Moooo" to the console.

CHICKEN FUNCTIONS

1. `def __init__(self, height, weight, age, name, eggsADay, isPregnant):` This function will be the constructor function of the chicken class.
2. `def chickenInfo(self):` This function will print out the info of this chicken in the following format:

```
height : 1.3 ft
weight : 13 lb
age : 2.7 years old
name : Lexi
eggsADay : 7 eggs
isPregnant : False
```

With the information relevant to that specific chicken.

3. `def cluck(self):` This function will print "Cluck cluck cluck" to the console.

HORSE FUNCTIONS

1. `def __init__(self, height, weight, age, name, maxSpeed, isRideable):` This function will be the constructor function of the horse class.

2. `def horseInfo(self)`: This function will print out the info of this horse in the following format:

```
height : 6.1 ft
weight : 392 lb
age : 12.2 years old
name : Steve
maxSpeed : 31.3 mph
isRideable : True
```

With the information relevant to that specific horse.

3. `def neigh(self)`: This function will print “Neigh” to the console.

DOG FUNCTIONS

1. `def __init__(self, height, weight, age, name, breed, isFriendly)`: This function will be the constructor function of the dog class.
2. `def dogInfo(self)`: This function will print out the info of this dog in the following format:

```
height : 2.8 ft
weight : 80 lb
age : 3.0 years old
name : Roger
breed : Golden Retriever
isFriendly : True
```

With the information relevant to that specific dog.

3. `def woof(self)`: This function will print “Bark bark” to the console.

6. Testing

No tester program will be provided for this assignment. Part of the purpose of this assignment is to provide students with the opportunity to self-test their own programs and create their own test cases.

7. Requirements

Your program must be named “barnyard.py.” The `__init` function must be used in all classes in this program

8. Help & Advice

If you feel overwhelmed by this assignment, don’t be. A good starting point is to first create skeleton definitions of the required functions that returns dummy values. From there, work on the function that you feel is the easiest to

implement. Once defined, run the functionsTester.py program and see if you were able to successfully implement it. From this point, rinse and repeat.

If you need any kind of clarification to anything on this assignment, such as if you're confused about the requirements for one of the functions, or if you encounter a bug error that you are unable to figure out on your own (this includes any potential errors arising from the soupTaster.py tester program), please send an email to chsprogvol@gmail.com. You may expect a response within 24 to 48 hours.