Object Oriented Programming In Python

Activity Sheet: Week 2

Task 1: Enhance the Person Class. (*The program is available as Class.py*)

Enhance the provided person class. You should try some of the following:

- Add some 'getter' methods to go with the existing methods that set attributes.
- Add some new attributes of the person that could be useful (*for what? – in week 1 we imagined an agency, but change this if you wish*).

Try out your additions with a simple program (start from person-ex1.py)

Task 2: Create a new class. You can invent a scenario, but here is a possible one:

You have so many pets at home you are getting confused. They all have names, of course. Some eat twice a day, some once and other less often. What they eat also varies: the dog was not happy with rabbit food! It is also useful to record their age, as they do get old.

- Think of a name for the class
- *(No need to define a constructor at first)*
- Suggest some attributes for the class and define methods to set them.
- Create a description of an object of the class.

Demonstrate the use of you class in a simple command line program. It should make use of multiple objects of the class and show how objects are updated.

Task 3 Add a Constructor to your Person.py: You should consider which attribute should be initialised from parameters and be sure to give sensible defaults to all the others.

- Check that all the methods of the class are sure to run without failing on a 'new' object.
- Modify the command line user program – does the default constructor still work?

Task 4: Add a Constructor to the Class from Task 2

Also add a constructor to the class you created in task 2. As before, ensue ll attributes ar initialised.

Homework Task: Care of Cats Project

This homework asks you to create a simple program consisting of a Class (in one file) and a user program (in a second file). The program creates a simple game: CareForCats.

A cat has a limited number of actions / interactions

- You can feed it and it eat
- You can pat it: if it is happy it will purrrr, but if it is hungry or tired it may bite or scratch you.
- You can tell it (???) to go to sleep: since cats like sleeping, it always does.

At any time, a cat is either 'awake' or 'asleep'. The rules for keeping cats are shown in the table below:

A Cat Can	When
Eat	When it is awake. Don't fed a sleeping cat.
Sleep	When it is awake. Don't try suggesting sleep when it already asleep.
Pat (i.e. be patted)	Anytime. If it is asleep, it wakes up. If it is hungry, it bites you; if it is tired it scratches you. Otherwise it purrrrrrs
be tired	If it has been awake for longer than it's last sleep
be hungry	If it has not been fed for a while

An example of the program running is shown below:

```
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 1
[P]at, [F]eed, [S]leep, [C]heck> p
Purrrrrrrr!
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 1
[P]at, [F]eed, [S]leep, [C]heck> p
Paws is tired. SCRATCHES you
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 1
[P]at, [F]eed, [S]leep, [C]heck> s
Paws goes to sleep
```

```
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 2
[P]at, [F]eed, [S]leep, [C]heck> p
Whiskers is tired. SCRATCHES you
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 2
[P]at, [F]eed, [S]leep, [C]heck> s
Whiskers goes to sleep
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 1
[P]at, [F]eed, [S]leep, [C]heck> c
Paws is asleep
```

```
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 1
[P]at, [F]eed, [S]leep, [C]heck> p
You have woken Paws
Paws is hungry. BITES you
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number> 2
[P]at, [F]eed, [S]leep, [C]heck> f
YOU ARE STUPID! Can't you see
Whiskers is asleep
Choose a cat:
  1: Paws
  2: Whiskers
  Cat number>
```

A starter program is provided for you to use if you wish (Files Cat.py, patFeedSleep.py). It is recommended that you work in stages:

- What attributes are needed in the cat class?
- What commands are available to the player?
- Add more features, such as adding new cats (kittens?)