CS 100
Homework 11

Do all of the items below and **submit** a text file created with the IDLE editor (or other editor) with the name *dog.py* via Canvas. If you run into a problem, post to Canvas describing where you ran into trouble or email your instructor or classroom assistant, or ask your question during recitation hours. If you know the answer to someone's question on Canvas, post a response. You get course credit for asking and answering questions in Canvas.

- Read Chapter 15 (Classes and objects) in the textbook.

- Read the Python tutorial section 9.3 (A First Look at Classes). The Python tutorial can be accessed through the documentation installed with IDLE:
  Help → Python Docs → Tutorial → 9.3. A First Look at Classes
  If you are using an alternate IDE, visit:
  https://docs.python.org/3/tutorial/classes.html#a-first-look-at-classes
  to browse the tutorial online.

In this homework you will design a class *Dog*. The file *dog.py* must begin with a comment containing your name, class and section, the posting date and number of the homework assignment.

Important Remark: When you modify the file *dog.py* the changes will not be reflected in the shell, even if you execute *import dog* again. Therefore, always start a new shell after modifying the class *Dog*.

Problem 1
Write a class definition line and a one line docstring for the class *Dog*. Write an *__init__* method for the class *Dog* that gives each dog its own *name* and *breed*. Test this on a successful creation of a *Dog* object.

```
>>> import dog
>>> sugar = dog.Dog('Sugar', 'border collie')
>>> sugar.name
Sugar
>>> sugar.breed
border collie
```

Problem 2
Add a data attribute *tricks* of type *list* to each *Dog* instance and initialize it in *__init__* to the empty list. The user does not have to supply a list of tricks when constructing a *Dog* instance. Make sure that you test this successfully.

```
>>> sugar.tricks
[]
```

Problem 3
Write a method *teach* as part of the class *Dog*. The method *teach* should add a passed string parameter to tricks and print a message that the dog knows the trick.

```
>>> sugar.teach('frisbee')
Sugar knows frisbee
```

Problem 4

Write a method *knows* as part of the class *Dog*. The method knows should check whether a passed string parameter is in the dog's list of tricks, print an appropriate message and return *True* or *False*.

```
>>> sugar.knows('frisbee')
Yes, Sugar knows frisbee
True
>>> sugar.knows('arithmetic')
No, Sugar doesn't know arithmetic
False
```

Problem 5

Create a class attribute *species* of type *str* to be shared by all instances of the class *Dog* and set its value to *'canis familiaris'*. The class attribute *species* should be defined within the class *Dog* but outside of any method.

```
>>> dog.Dog.species
'Canis familiaris'
>>> sugar.species
'Canis familiaris'
```