Pratice: Exam 1 Coding

You will have to hand-write code during lecture for the summative coding assessments in this course (the coding exams). Below are some example problems from each module.

Mod 01 - Python Foundations

Write a function `has_same_letters(word1, word2)` that returns `True` if every letter in `word1` also appears in `word2` and vice-versa. Note that they do not have to have the same *count* of each letter, as long as they have the same letters.

Examples

```
>>> has_same_letters("reheat", "theater")
True
>>> has_same_letters("reheat", "there")
False
```

Work

Code:

Running time: ____________________________

Mod 02 - Object-Oriented Programming

Write code to implement the following class diagram. Make sure to add `init` methods with the appropriate parameters where appropriate.

Examples

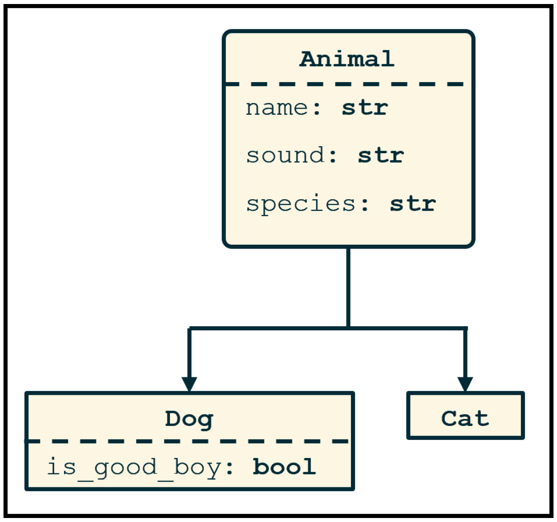

Figure 1: Class diagram to be implemented. Note that `name`, `sound`, `species`, and `is_good_boy` are instance variables.

Work

Code:

Mod 03 - Running Time Analysis and Test-Driven Development

Write a function `time_func(func, *args)` that takes as input a function and a tuple of arguments, runs that function 10 times, and returns the *minimum time requried* for function to complete.

Examples

```
>>> def double(x):
        return x*2

>>> time_func(double, ('hello'))
0.00005
```

Work

Code:

Write a suite of unittests for the Stack ADT. You can assume the stack provides typical methods for `push`, `pop`, `len`, and `peek`, and that it should raise an `IndexError` if you try to pop from an empty stack.

```
from stack import Stack
import unittest



unittest.main()
```

Mod 04 - Linear ADTs and Data Structures

Implement `add_last` in the `LinkedList` below. Note that you should *only* add code to the method `add_last` - do not create any other methods or add any other attributes to the class.

Work

Code:

```python
class Node:
    def __init__(self, data, link):
        self.data = data
        self.link = link

class LinkedList:
    def __init__(self):
        self._head = None
        self._len = 0

    def __len__(self):
        return self._len

    def add_last(self, data):
        """Your work here"""
```

Running Time: ______________________________