Bookmar

- Overview
- Entrance Survey
- Week 1
- Week 2
- Week 3
- Week 4
- Quiz
- Week 5
- Week 6
- Week 7
- Week 8
- Exit Survey
- Final Exam

Final Exam
Final due Mar 15, 2016 at 23:30 UTC

- Sandbox

Bookmark

Problem 5

In this problem, you will implement a class according to the specifications in the template file usresident.py. The file contains a `Person` class similar to what you have seen in lecture and a `USResident` class (a subclass of `Person`). `Person` is already implemented for you and you will have to implement two methods of `USResident`.

For example, the following code:

```
a = USResident('Tim Beaver', 'citizen')
print a.getStatus()
b = USResident('Tim Horton', 'non-resident')
```

will print out:

```
citizen
## will show that a ValueError was raised at a particular line
```

usresident.py

```
## DO NOT MODIFY THE IMPLEMENTATION OF THE Person CLASS ##
class Person(object):
    def __init__(self, name):
        #create a person with name name
        self.name = name
        try:
            firstBlank = name.rindex(' ')
            self.lastName = name[firstBlank+1:]
        except:
            self.lastName = name
        self.age = None
    def getLastName(self):
        #return self's last name
        return self.lastName
    def setAge(self, age):
        #assumes age is an int greater than 0
        #sets self's age to age (in years)
```

```
        self.age = age
    def getAge(self):
        #assumes that self's age has been set
        #returns self's current age in years
        if self.age == None:
            raise ValueError
        return self.age
    def __lt__(self, other):
        #return True if self's name is lexicographically less
        #than other's name, and False otherwise
        if self.lastName == other.lastName:
            return self.name < other.name
        return self.lastName < other.lastName
    def __str__(self):
        #return self's name
        return self.name

class USResident(Person):
    """
    A Person who resides in the US.
    """
    def __init__(self, name, status):
        """
        Initializes a Person object. A USResident object
inherits
        from Person and has one additional attribute:
        status: a string, one of "citizen", "legal_resident",
"illegal_resident"
        Raises a ValueError if status is not one of those 3
strings
        """
        # Write your code here

    def getStatus(self):
        """
        Returns the status
        """
        # Write your code here
```

Paste only your implementation of the `USResident` class in the box below. Do not leave any debugging print statements.

For this question, you will not be able to see the test cases we run. This problem will test your ability to come up with your own test cases.

```
1 # Paste here
```

Unanswered

You have used 0 of 10 submissions