DEPAUL UNIVERSITY
COLLEGE OF COMPUTING AND DIGITAL MEDIA

CSC 242 NOTES
Fall 2024/2025: Anthony Zoko

Week of Monday, October 6th, 2024

In Class Exercise:

Let's work on a few problems together. You will find the Magic8Ball template in the class zip file.

1. Implement a class **Magic8Ball** (that is a subclass of the object class) representing an implementation of the Magic 8 Ball. The Magic 8 Ball has 20 possible answers to a question. You can find the history and a list of the 20 questions here: https://en.wikipedia.org/wiki/Magic_8-Ball . You need to implement a ball with supporting six methods:
    a. **__init__**() which either takes no parameters or takes a list representing answers. If nothing is passed in the constructor, the original 20 answers will be used. Otherwise, you can pass in a different list of answers. I recommend you set default list as empty and then populate it in the constructor if the list is empty.
    b. **get**() which returns the current answer.
    c. **shake**() which simulates the shaking of the 8 ball to get a random answer. Only shake will change the answer. In other words, after you **shake**() the ball, get will return the same answer until a new shake. Note: Per the Wikipedia article the intention was to turn the ball but not shake. We'll keep using the shake idea 😊
    d. **numAnswers**() which returns the number of answers in the ball..
    e. **__str__**() which returns a friendly string with information about how many answers it contains.
    f. **__iter__():** Allows you to iterate through all the answers in the Magic8Ball.

The following shows how the **Magic8Ball** class and its methods could be used:

```
>>> b=Magic8Ball()
>>> b.get()
'Cannot predict now'
>>> b.get()
'Cannot predict now'
>>> b.shake()
>>> b.get()
'Outlook good'
>>> b.get()
'Outlook good'
>>> b.get()
'Outlook good'
>>> b.numAnswers()
20
>>> lst=['Answer 1','Answer 2','Answer 3']
>>> b2=Magic8Ball(lst)
>>> b2.get()
'Answer 2'
>>> b2.get()
'Answer 2'
>>> b2.shake()
>>> b2.get()
'Answer 3'
>>> b2.get()
'Answer 3'
>>> str(b2)
'I am a Magic 8 Ball with 3 answers'
>>>
```

```
>>> b3 = Magic8Ball()
>>> for answers in b3:
        print(answers)


It is certain
It is decidedly so
Without a doubt
Yes definitely
You may rely on it
As I see it, yes
Most likely
Outlook good
Yes
Signs point to yes
Reply hazy try again
Ask again later
Better not tell you now
Cannot predict now
Concentrate and ask again
Don't count on it
My reply is no
My sources say no
Outlook not so good
Very doubtful
>>>
```

. Implement the CookieJar object as shown in the screenshots. ***You should not change the Cookie implementation***. Since the Cookie object picks a random type of cookie if a cookie type is not specified, your results may be slightly different than mine below. The docstrings in the template contain descriptions of each method.

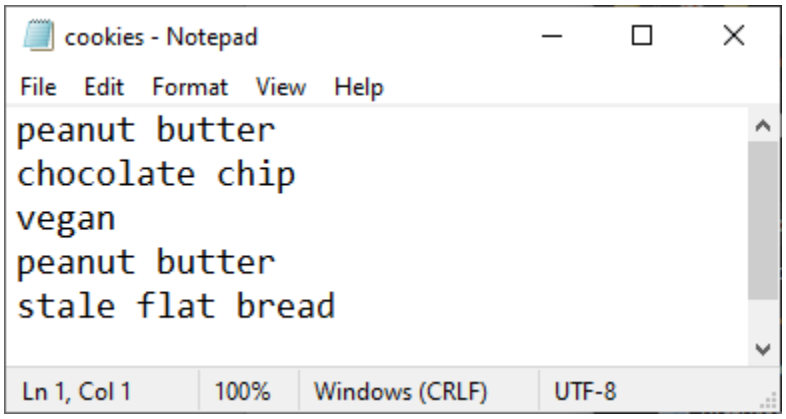
Grading Rubric:

- 10 Points: Proper reading / writing to a file
- 10 points: getAllCookieCount, getCookieCount
- 10 points: Constructor, getCount, addCookie, getCookies
- 5 points: Iterator operator
- 5 points: Contains operator

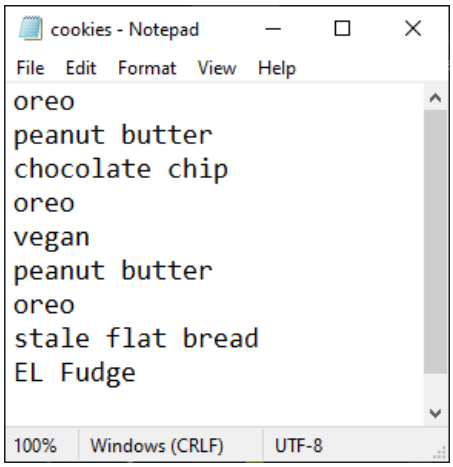
```
>>> jar=CookieJar()
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie())
>>> jar.addCookie(Cookie('stale flat bread'))
>>> jar.getCount()
5
>>> jar.getCookies()
[Cookie('peanut butter'), Cookie('chocolate chip'), Cookie('vegan'), Cookie('peanut butter'), Cookie('stale flat bre
>>> jar.getCookieCount('vegan')
1
>>> jar.getCookieCount('oreo')
0
>>> jar.getCookieCount('stale flat bread')
1
>>> jar.getAllCookieCount()
{'peanut butter': 2, 'chocolate chip': 1, 'vegan': 1, 'stale flat bread': 1}
>>> 'oreo' in jar
False
>>> 'vegan' in jar
True
>>> for cookie in jar:
        print(cookie)


peanut butter
chocolate chip
vegan
peanut butter
stale flat bread
```

```
>>> jar.saveCookies()
True
```



If I change the contents on cookie.txt to the following and load a new jar:




```
>>> jar=CookieJar()
>>> jar.loadCookies()
True
>>> jar.getAllCookieCount()
{'oreo': 3, 'peanut butter': 2, 'chocolate chip': 1, 'vegan': 1, 'stale flat bread': 1, 'EL Fudge': 1}
>>> for cookie in jar:
        print(cookie)


oreo
peanut butter
chocolate chip
oreo
vegan
peanut butter
oreo
stale flat bread
EL Fudge
```