

Homework #3: Magic Eight Ball

Using a magic eight ball the user can:

- Ask a question
- Roll the ball
- Receive an answer (prediction about whether the question will come to be)
- **Example:**
 - Magic Eight Ball prompts the user to ask a question
 - User asks: “Will I get an A on HW1 for SI206?”
 - Magic Eight Ball gives one of eight possible answers (listed below)
 - Magic Eight Ball continues to ask for the next question until the user ends the game



Instructions

For this assignment, you will be writing the *MagicEightBall* class with the following methods:

- An **`__init__(self, answers)`** method: This will initialize a new *MagicEightBall* class
 - Set the attribute **`answers_list`** to the **`answers`** argument. This is a list of the eight possible answers a player could receive.
 - Set the attribute **`question_history_list`** to an empty list.
 - Set the attribute **`answers_history_list`** to an empty list.
- A **`__str__(self)`** method: Returns a string with all of the answers in **`question_history_list`** separated by commas.
 - If no questions have been asked yet, return an empty string
- A **`get_fortune(self, question)`** method:
 - Checks if the question has been asked before
 - If it has, this method returns “I’ve already answered this question”
 - If the question has not been asked before, pick an answer at random from **`answers_list`**.
 - Add the index of the answer in **`answers_list`** to **`answers_history_list`**

- e.g. if **answers_list** is ['yes', 'no'] and the answer is 'yes', you should add 0 to **answers_history_list**
 - Returns the answer
- A **play_game(self)** method: This method controls the game play for the **MagicEightBall** object
 - Prompts the user to ask a question: "Please enter a question: "
 - If the question is "I'm done playing" then print "Goodbye" and end the game
 - Otherwise, add the question to **questions_history_list** and use the **get_fortune** method to generate a fortune
 - Print the fortune
 - Add the question to **questions_history_list**
 - Prompts the user to ask the next question: "Please enter the next question: "
- A **print_answer_frequencies(self)** method: This method prints out the answers
 - Using the **answers_history_list**, count how many times each answer is given.
 - Print out "The answer '<answer>' has been given <number> of times."
 - **Hint:** You can use the .count() method
 - **Hint:** "I've already answered this question" should not appear in answers_history_list
 - Returns a dictionary that maps answers to their frequency
 - If there are no answers in **answers_history_list**, it will print "None yet" and return an empty dictionary
- A **main()** function:
 - Defines the possible answers into a list: Definitely, Most Likely, It is certain, Maybe, Cannot predict now, Very doubtful, Don't count on it, Absolutely not
 - Create the **MagicEightBall** object
 - Initiate the game play using the **play_game()** method
 - Shows the output of **print_answer_frequencies()**

Sample output from the main method:

```
Please enter a question: will it snow today?
Most Likely

Please enter the next question: should I bring my gloves with me?
Don't count on it

Please enter the next question: will it snow today?
I've already answered this question

Please enter the next question: should I study in the ugli?
Maybe

Please enter the next question: I'm done playing
Goodbye
The answer 'Most Likely' has been given 1 times
The answer 'Don't count on it' has been given 1 times
The answer 'Maybe' has been given 1 times
```

Grading Rubric - Total of 60 Points

- 5 points: the `__init__` method sets the object's ***answers_list***, ***questions_history_list***, and ***answers_history_list*** correctly to the passed arguments, sets both the object's ***questions_history_list*** and ***answers_history_list*** attributes to an empty list
- 5 points: the `__str__` method returns a string with all answers in ***answers_list*** separated by commas
 - Correct answers for a list `["Definitely", "Most likely", "It is certain", "Maybe", "Cannot predict now", "Very doubtful", "Don't count on it", "Absolutely not"]`
- 5 points: the ***get_fortune*** method returns "I've already answered this question" if the question has already been asked
- 5 points: the ***get_fortune*** method adds the answer to ***answers_history_list***
 - If a question has already been asked, ***get_fortune*** does not add the "I've already answered this question" to ***answers_history_list***
- 5 points: the ***play_game*** method continually prompts the user for a question, using prompt "Please enter a question" as long as they don't input "I'm done playing"
- 5 points: the ***play_game*** method adds the questions to ***questions_history_list***
- 5 points: the ***play_game*** method uses the ***get_fortune()*** method to correctly get the answer

- 10 points: ***print_answer_frequencies*** prints “The answer ‘<answer>’ has been given <number> of times.” for each of the answers from ***answers_history_list*** on separate lines
- 3 points: ***print_answer_frequencies*** returns “None yet” if there are no answers in ***answers_history_list***
- 3 points: ***answers_list*** is properly defined and used in the ***main()*** function
- 3 points: the ***MagicEightBall*** object is properly defined and used in the ***main()*** function
- 3 points: the ***play_game*** method is used correctly in the ***main()*** function
- 3 points: the ***print_answer_frequencies*** method is used correctly in the ***main()*** function

Extra Credit: 6 points

Create a ***my_test()*** function that creates a ***MagicEightBall*** object and tests each of the possible outcomes.

- **1 point:** Correct output from ***print_answer_frequencies*** when no questions have been asked.
- **2 point:** Correct behavior from ***print_answer_frequencies*** when ***answers_list*** is ['It is certain', 'It is certain', 'Don't count on it'] and ***answers_history_list*** is [0, 1, 1]
 - **Hint:** you can modify the value of attributes on a class that's already been created. For example, if your ***MagicEightBall*** object is called ***eight_ball***, you can make ***answer_history_list*** equal to an empty list by setting ***eight_ball.answer_history_list = []***
- **1 point:** Correct output from ***get_fortune*** when a question has already been asked.
- **1 point:** Correct output from ***play_game*** when the first question asked is “I’m done playing.”

Running Your Code:

If you are having trouble running your code / interacting with the program in VSCode, click the arrow in the top right corner of your VSCode window. Then, hit “Run Python File.”

