

# CISC 110 Assign 6: Magic 8-Ball

You will complete this assignment during Lab 7, but don't worry if you don't finish all of the steps. This assignment is purposely challenging to be good practice for the second test. The most important thing is to ask questions of your TA and professor until you understand defining functions. This is the last topic of the course that will be on the second test.

In this assignment, you will complete the Magic 8-Ball app you started in Lab 6. The primary purpose of the assignment is to practice writing functions. First you will write a function that randomly chooses a message from your array of strings; you have already done this without writing a function, so the only purpose of the change will be to practice writing a function that returns a result. Then you will write a function that checks whether an answer has already been used. You will keep selecting random answers from the answer array until it chooses a new one, so it doesn't keep repeating itself. You will also add a second text field to the 8-Ball to display the big 8 at the start and while the ball is being dragged, and you may improve your graphic of the 8-ball and add any text and background that you wish.

Below are the main steps for your assignment.

1. Add a second dynamic text field in the 8-ball Movie Clip that displays a large 8 at the beginning of your movie and when the ball is being dragged.
2. Write the definition for a function that returns a random answer from your answer array. Move the code that already does this inside of your new function definition and modify it to return the random answer, using a return statement.

Here is the header line for your function definition:

```
function randomAnswer( ): String
```

3. Test that your function works correctly by placing the following function calls and trace statements inside of your constructor function. This is not part of the Magic 8-Ball game. The idea is to try out a function definition to ensure it's correct before you use it in a game or other app. These lines could be removed once you've implemented your whole game. They are only included for testing purposes. They also show you how to use your randomAnswer function, i.e., how to write a function call for a function that returns a result. The result of the function must be assigned to a variable or displayed.

```
var ans = randomAnswer( );  
trace( "Answer is: " + ans );  
ans = randomAnswer( );  
trace( "Answer is: " + ans );
```

4. Use this function in your game by calling it inside of your stop drag handler function. Now your Magic 8-Ball should work again as it did before you wrote the function.
5. Save your Magic 8-Ball game in a separate file with a different name before you start the next steps, since they are more difficult. If you don't complete those steps, you can show a working program to your TA and receive most of the marks. In the next steps, you will write a function that checks whether an answer has already been used. You will keep selecting random answers from the answer array until it chooses a new one, so that it doesn't keep repeating itself.
6. Define another global array that also contains strings as its elements, right after the definition of your answer array, and call it `usedList`. However, don't put any strings in it when you define it. Therefore, `usedList` starts off as an empty array.
7. Modify your stop drag handler function so that each time you choose and display a random answer in the 8-ball, that answer is added to the end of `usedList`, using the `push` method.
8. Write the definition for a function that checks whether an answer has already been used by checking whether that answer is in the `usedList` array. Your function should return `true` if the answer is in `usedList` and `false` otherwise. Use a `for` loop to go through `usedList` and check if each element is equal to `answer`.

Here is the header for your function, along with explanatory comments. It's good style to include explanatory comments for every function definition.

```
// Check whether answer has already been used
// Returns true if answer is in usedList and false if not
function usedAnswer( answer: String ): Boolean
```

9. Test that your function works correctly by placing the statements below inside of your constructor function. These lines could be removed once you've implemented your whole game. They are only included for testing purposes and to show how to call the function. The first two lines place the strings "Signs point to yes" and "Outlook is good" in the previously empty `usedList` array, so the only elements in the array are these two strings. The two trace statements call the `usedAnswer` function. If your function definition is correct, the first trace statement should print `false`, since "Ask again later" is not in the `usedList` array, and the second trace statement should print `true`, since "Outlook is good" is in the `usedList` array. Be sure you understand what these lines are doing before you go on to the last step.

```
usedList[0] = "Signs point to yes";
usedList[1] = "Outlook is good";
ans = "Ask again later";
trace( usedAnswer( ans ) );
ans = "Outlook is good";
trace( usedAnswer( ans ) );
```

10. Use this function in your game to choose random answers from the answer array until one is found that hasn't already been used. Modify your stop drag handler function. Using a while loop, repeatedly call the `randomAnswer` function to obtain a random answer and then check if the answer is in `usedList` using the `usedAnswer` function until you get a random answer that is not in `usedList`.

If you finished all of these steps, CONGRATULATIONS!

You don't need to publish this assignment on the CISC 110 website. Just show your work to your TA.

## Assignment 6 Marking Scheme (2% of final mark)

**Marked out of 10:**

**1 mark:** Second dynamic text field in the 8-ball that displays a large 8 only at the beginning and when the ball is being dragged

**3 marks:** Definition of the `randomAnswer` function

**1 mark:** Testing of the `randomAnswer` function shows function is correct

**1 mark:** Uses the `randomAnswer` function correctly in the stop drag handler function

**1 mark:** Definition of a second global array of strings, called `usedList`, this one empty, right after the definition of the first array

**1 mark:** Each time a random answer is chosen and displayed in the 8-Ball, it's added to the end of `UsedList`.

**1 mark:** Definition of function `usedAnswer`

**1 mark:** Repeatedly chooses a random answer from the answer array until one is found that hasn't already been used.