Loops Assignments

1. Creating a Lesson Plan(Using `for` loop)

Objective: Create a list of lesson objects in TypeScript with alternating statuses to indicate whether each lesson is running this year.

Steps to Follow:

1. **Create a blank array:** Start by setting up an empty array named `myWork` that will hold objects.
2. **Use a loop to create lessons:** Write a `for` loop that runs from 1 to 10. In each iteration:
    - Create an object representing a lesson.
    - Each lesson should have a `name` property, which is a string like "Lesson 1", "Lesson 2", etc.
    - Each lesson should also have a `status` property, which alternates between `true` and `false`. This means "Lesson 1" will have `status: true`, "Lesson 2" will have `status: false`, and so on.
3. **Add the lesson to the array:** Push each lesson object into the `myWork` array.
4. **Print the result:** Finally, log the `myWork` array to the console to see the list of lessons.

Hints:

- Use a ternary operator to set the `status` property based on whether the lesson number is odd or even.
- The modulo operator `%` can help determine if a number is even or odd.

2. Guessing Game (Using `while` loop)

Objective: Create a simple number guessing game where the user tries to guess a randomly generated number between 1 and a specified maximum value using a predefined set of guesses.

Steps to Follow:

1. **Set a maximum value:** Create a variable to store the maximum value for the number guessing game.
2. **Generate a random number:** Use `Math.random()` and `Math.floor()` to generate a random number between 1 and the maximum value. Log this value to the console for development purposes.
3. **Track the guess status:** Create a variable to track whether the user's guess is correct. Set it to `false` initially.
4. **Simulate user guesses:** Use an array to store a series of predefined guesses.
5. **Iterate over guesses:** Use a `while` loop to iterate over the predefined guesses until the correct guess is made.

6. **Check the user's guess:** Inside the loop, check if the current guess matches the random number. Provide feedback if the guess is too high or too low.

3. Counter Incrementer (Using `do while` loop)

Objective: Create a program that increments a counter by a specified step value using a `do...while` loop and prints the counter value to the console until it reaches or exceeds 100.

Steps to Follow:

1. **Set the starting counter to 0:** Create a variable `counter` and initialize it to 0.
2. **Create a variable, `step`, to increase your counter by:** Define a variable `step` to hold the value by which the counter will be incremented.
3. **Add a `do...while` loop:** In the loop, print the counter to the console and increment it by the step amount each iteration.
4. **Continue to loop until the counter is equal to or more than 100:** The loop should run as long as the counter is less than 100.

4.Exploring Objects with `for...in` Loop

Objective: Practice working with objects in TypeScript and iterating over their properties using a `for...in` loop.

Instructions:

1. **Create a simple object with three items:**
   - Define an object called `myObject` with three properties: `item1`, `item2`, and `item3`, each with corresponding string values.
2. **Use a `for...in` loop to get properties' names and values from the object:**
   - Iterate through the properties of `myObject` using a `for...in` loop.
   - Inside the loop, print each property's name and its corresponding value to the console.

5.Exploring Arrays with Loops(Using loop)

Objective: Practice working with arrays in TypeScript and using `for` loops and `for...of` loops to iterate through array elements.

Instructions:

1. **Create an empty array:**
   - Define an empty array called `myArray`.
2. **Run a loop 10 times, adding a new incrementing value to the array:**
   - Use a `for` loop to iterate 10 times.

- In each iteration, add a new incrementing value (starting from 1) to the `myArray`.

3. **Log the array into the console:**
   - After populating the array, log the `myArray` into the console.

4. **Use the `for` loop to iterate through the array:**
   - Use a `for` loop to iterate through the array elements.
   - Adjust the number of iterations based on the number of values in the array.
   - Output each array element along with its index into the console.

Use the `for...of` loop to output the value into the console from the array:

- Use a `for...of` loop to iterate through the array elements.
- Output each array element directly into the console.