Lab 1:

1. Create an HTML file named lab1.html and embed JavaScript using a <script> tag.
2. Write a **single-line comment** that briefly explains what JavaScript is (e.g., “JavaScript is used to make web pages interactive”).
3. Write a **multi-line comment** describing common uses of JavaScript (e.g., form validation, animations, dynamic content).
4. Use console.log() to display:
   - Your **name** (e.g., "Ali ahmd"),
   - Your **age** (e.g., 22),
   - Your **favorite programming language** (e.g., JavaScript).
5. Add a comment above each console.log() statement explaining what it does.

Lab 2:

1. Declare variables to store personal information:
   - firstName (e.g., "Ali"),
   - lastName (e.g., "ahmed"),
   - age (e.g., 22),
   - isStudent (e.g., true).
2. Use different naming styles:
   - camelCase (e.g., firstName),
   - PascalCase (e.g., LastName),
   - underscore (e.g., age_in_years).
3. Try creating invalid variable names such as:
   - Starting with a number (e.g., 123name),
   - Using special characters (e.g., my-name),
   - Using reserved keywords (e.g., let).
4. Log all valid variables to the console.
5. Use try and catch to hand the error

Lab 3:

1. Ask the user to enter a numerical grade (e.g., 85).
2. Use if, else if, and else to assign a letter grade:
    - 90–100: "Excellent"
    - 75–89: "Good"
    - 60–74: "Pass"
    - Below 60: "Fail"
3. Display the result using a message.

Lab 4:

1. Generate a random price (e.g., between 10 and 100).
2. Round it to two decimal places using formatting methods.

Lab 5:

1. Ask the user to enter a product name (e.g., "Laptop").
2. Check if the input includes certain keywords using includes(), indexOf(), or startsWith().
3. Convert input to lowercase for case-insensitive comparison.

Lab 6:

1. Get the current date.
2. Add 3 days to the current date.

Lab 7:

1. Simulate a login system that allows up to 3 attempts.
2. Use a while loop to keep asking for a password until it’s correct or attempts run out.
3. Use a do...while loop to ask for a password at least once.

Lab 8:

1. Loop through numbers 1 to 50.
2. Use continue to skip even numbers and only print odd ones.

3. Use break to stop the loop when a specific number is reached (e.g., 37).

Lab 9:

1. Ask the user for a number (e.g., 4).
2. Use nested loops to print a right-angled triangle of stars:
   - Line 1: *
   - Line 2: **
   - Line 3: ***
   - Line 4: ****

Lab 10:

1. Ask the user for an initial amount (e.g., 1000).
2. Apply a tax rate and discount using +=, -=, *=, /=.
3. Calculate the final price after all operations.

Lab 11:

Objective: Validate password strength using conditions.

1. Ask the user to enter a password.
2. Check if the password:
   - Is at least 8 characters long,
   - Contains both letters and numbers.
3. Display whether the password is strong or weak.

Lab 12: Random Promo Code Generator

Objective: Generate random codes using loops and random numbers.

1. Create a loop that generates 5 promo codes.
2. Each code should start with a prefix (e.g., "PROMO") followed by a random 4-digit number.
3. Display all generated codes.

Lab 13:

1. Wrap a function in an IIFE that runs immediately when the page loads.
2. Inside the IIFE, define a variable (e.g., config) and log it.
3. Ensure that the variable is not accessible outside the IIFE.

Lab 14:

1. Simulate a calculator that throws an error if the user enters invalid input (e.g., dividing by zero).
2. Catch the error and display a user-friendly message.
3. Use finally to log that the operation has completed.