Day 4 & 5 : Assignments

1. Using Promises and try...catch
    a. Create a function fetchData that returns a Promise.
    b. Use fetch to get data from an API endpoint (e.g., JSONPlaceholder).
    c. If the fetch is successful, resolve the Promise with the data.
    d. If there's an error (network issue, invalid URL, etc.), reject the Promise with an appropriate error message.
    e. Call fetchData and handle the resolved data and errors using then, catch, and finally.

2. Advanced Closure with Configuration
    a. Create a function createCounter that takes an initial value and step size as parameters.
    b. The function should return an object with methods: increment, decrement, reset, and getValue.
    c. The increment method should increase the counter by the step size.
    d. The decrement method should decrease the counter by the step size.
    e. The reset method should reset the counter to the initial value.
    f. The getValue method should return the current value of the counter.
    g. Demonstrate the closure by creating a counter with an initial value of 10 and step size of 5.

3. Using bind Method with Event Listeners
    a. Create a class User with a constructor that takes a name parameter.
    b. The class should have a method greet that logs a greeting message including the user's name.
    c. Use the bind method to bind greet to the current instance within the constructor.
    d. Demonstrate the class by creating an instance and calling the greet method.

4. Combining Promises and Closures
    a. Create a function createDataFetcher that returns an object with two methods: fetchData and getData.
    b. The fetchData method should fetch data from an API and store it in a closure variable.
    c. The getData method should return the fetched data.
    d. Demonstrate using the fetchData and getData methods.

5. Handling Asynchronous Operations with Error Handling
    a. Create a function fetchDataWithRetry that takes a URL and a maximum retry count as parameters.
    b. Use a closure to maintain the retry count and a private function fetchData that returns a Promise for fetching data from the URL.
    c. Implement the fetchData function with error handling to retry fetching the data up to the maximum retry count.
    d. Demonstrate the function by calling fetchDataWithRetry with a URL and handling the resolved data or errors.