**Medium to Hard Coding Questions:**

1. **Number and Math:**
   * **Question:** Create a function formatCurrency(amount, currencyCode) that takes a numerical amount and a currencyCode (e.g., 'USD', 'EUR', 'INR'). The function should return a formatted string representing the currency. Implement logic to handle different currency formats (e.g., placing the symbol before or after the amount, using commas for thousands separators, and handling decimal places appropriately). You might need to research the common formatting conventions for different currencies.
2. **Strings:**
   * **Question:** Write a function findLongestPalindrome(str) that takes a string str as input and returns the longest palindromic substring within it. A palindrome is a string that reads the same forwards and backward (e.g., "racecar", "madam"). This will require you to efficiently check for palindromes within the string.
3. **HTML, CSS Review & console.log (Conceptual):**
   * **Question:** Imagine you have a simple HTML list (<ul> with <li> items). Describe how you would use JavaScript and console.log statements to:
     + Select all the list items.
     + Iterate through each list item and log its text content to the console.
     + Add a class "highlight" to every second list item.
     + Log the total number of list items to the console.
   * *(This is less about writing a complex function and more about demonstrating your understanding of DOM manipulation and using the console for debugging.)*
4. **Variables:**
   * **Question:** Explain the concepts of variable scope (global, local, block scope) in JavaScript. Provide code examples to illustrate each type of scope and discuss potential issues that can arise from misunderstanding scope (e.g., variable shadowing).
5. **Booleans and If-Statements:**
   * **Question:** Create a function evaluateConditions(obj) that takes an object obj with various properties. The function should evaluate a series of complex conditions based on the values of these properties and return a specific string based on the outcome. For example:
     + If obj.age is greater than 18 AND obj.hasLicense is true, return "Eligible to drive".
     + If obj.temperature is above 30 OR obj.humidity is above 70, return "Warning: High discomfort level".
     + If obj.role is either "admin" OR "editor" AND obj.isActive is true, return "Access granted".
     + Include at least 3-4 more complex conditions using different logical operators (&&, ||, !).
6. **Functions:**
   * **Question:** Write a function createCalculator() that returns an object. This object should have four methods: add(a, b), subtract(a, b), multiply(a, b), and divide(a, b). Demonstrate how you would use this returned object to perform calculations. This exercise tests your understanding of closures and how functions can create and return objects with encapsulated behavior.
7. **Objects:**
   * **Question:** Design a system to represent a library. Create JavaScript objects for Book and Library.
     + The Book object should have properties like title, author, ISBN, and isBorrowed (boolean). It should also have a method borrow() that sets isBorrowed to true and a method returnBook() that sets it to false.
     + The Library object should have a property books (an array of Book objects) and methods like addBook(book), findBookByISBN(isbn), listAvailableBooks(), and listBorrowedBooks(). Implement these methods.
8. **Document Object Model (DOM):**
   * **Question:** Imagine you have the following HTML structure:  
     HTML  
     <div id="container">  
      <h1>My List</h1>  
      <ul id="myList">  
      <li class="item">Item 1</li>  
      <li class="item">Item 2</li>  
      <li class="item">Item 3</li>  
      </ul>  
      <button id="addButton">Add Item</button>  
     </div>  
     Write JavaScript code that does the following:
     + Adds a new list item with the text "New Item" to the end of the <ul> when the "Add Item" button is clicked.
     + When any list item is clicked, it should be removed from the list.
9. **HTML, CSS, and JavaScript Together:**
   * **Question:** Create a simple interactive counter.
     + **HTML:** Include a display area (e.g., a <span> or <div>) to show the current count and two buttons: "Increment" and "Decrement".
     + **CSS:** Style the counter elements (display, buttons) to be visually appealing.
     + **JavaScript:**
       - Initialize a counter variable to 0.
       - When the "Increment" button is clicked, increase the counter and update the display.
       - When the "Decrement" button is clicked, decrease the counter and update the display.
       - Ensure the counter doesn't go below 0.
10. **Arrays and Loops:**
    * **Question:** Write a function processTransactions(transactions) that takes an array of transaction objects. Each transaction object has properties like type ('deposit' or 'withdrawal') and amount. The function should:
      + Calculate the total balance after processing all transactions (start with a balance of 0).
      + Return an object containing the finalBalance, the total amount of all deposits, and the total amount of all withdrawals. Use loops (e.g., for, for...of) to iterate through the transactions.
11. **Arrays and Loops (Part 2):**
    * **Question:** Implement a function groupByCategory(items, categoryKey) that takes an array of objects items and a string categoryKey. The function should group the items based on the value of the categoryKey property and return an object where the keys are the unique category values and the values are arrays of the items belonging to that category.
      + Example:  
        JavaScript  
        const products = [  
         { name: 'Laptop', category: 'Electronics', price: 1200 },  
         { name: 'T-Shirt', category: 'Apparel', price: 25 },  
         { name: 'Mouse', category: 'Electronics', price: 30 },  
         { name: 'Jeans', category: 'Apparel', price: 70 },  
        ];  
        const grouped = groupByCategory(products, 'category');  
        // Expected output:  
        // {  
        // Electronics: [ { name: 'Laptop', category: 'Electronics', price: 1200 }, { name: 'Mouse', category: 'Electronics', price: 30 } ],  
        // Apparel: [ { name: 'T-Shirt', category: 'Apparel', price: 25 }, { name: 'Jeans', category: 'Apparel', price: 70 } ]  
        // }

By tackling these questions, you'll be applying the fundamental JavaScript concepts in more complex scenarios, which will significantly enhance your understanding and problem-solving skills. Good luck!