## Part 1: Beginner Questions

1. **Create and Access:** Create an object named `book` with three properties: `title`, `author`, and `year`. Assign a string, a string, and a number as their respective values. Then, log the `title` using dot notation and the `year` using bracket notation.
2. **Add and Update:** Start with an object named `car` with a `make` and `model`. Add a `color` property to it, then update the `make` property to a new value. Log the entire `car` object to see the changes.
3. **Delete a Property:** Create an object named `userProfile` with `username`, `email`, and `isVerified` properties. Delete the `isVerified` property and then log the `userProfile` object to confirm the deletion.
4. **Property Existence:** Create a `person` object with a `name` and `age`. Use the **`in`** operator and the **`hasOwnProperty`** method to check if the `person` object contains the `name` property. Log the boolean result for both checks.
5. **Method Basics:** Create an object named `calculator` with a property `value` set to 0. Add two methods to it: `add` and `subtract`. The `add` method should take a number and add it to the `value`, and the `subtract` method should take a number and subtract it. Call these methods and log the `value` after each operation.
6. **Dot vs. Bracket Notation:** Declare a variable `propertyName` with the string value `'price'`. Create a `product` object with an `id` and a `price`. Log the product's `price` using the `propertyName` variable and bracket notation.
7. **Simple Array of Objects:** Create an array named `tasks`. Add two objects to this array, each with `id`, `description`, and `isCompleted` properties. Log the `description` of the first task using dot and bracket notation.
8. **Finding an Item:** Using the `tasks` array from the previous question, write a function that takes a task `id` and returns the corresponding task object. If the task is not found, return `null`.

---

## Part 2: Intermediate Questions

9. **Iterating with `for...in`:** Create an object named `studentGrades` with subject names as keys and scores as values (e.g., `{ math: 90, science: 85, history: 78 }`). Use a **`for...in`** loop to iterate over the object and log each key-value pair to the console.
10. **Array of Objects Iteration:** Using the `students` array from your provided example, use a **`for`** loop to iterate through the array and log the `name` of each student.

11. **Method with `this`:** Create a `user` object with `name`, `age`, and a `canVote()` method. The `canVote` method should use `this.age` to return `true` if the user's age is 18 or older, and `false` otherwise.
12. **Destructuring:** Create a `settings` object with properties `theme`, `fontSize`, and `notifications`. Use object destructuring to extract `theme` and `fontSize` into separate variables. Log both variables.
13. **Destructuring with Renaming and Defaults:** Create a `userPrefs` object with a `viewMode` property. Use destructuring to extract `viewMode` and rename it to `displayMode`. Also, provide a default value for a `language` property that doesn't exist in the object. Log both variables.
14. **`Object.keys/values/entries`:** Using the `studentGrades` object from question 9, use **`Object.keys()`** to log all the subject names. Then, use **`Object.values()`** to log all the scores.
15. **Complex Iteration and Calculation:** Using the `students` array from your provided example, write a function that iterates through the array and calculates the average score for each student. Log each student's name along with their calculated average score.