Object , Array and function

Assignment 1: Building Your Friend List

Learning Objective: Practice working with objects and arrays in TypeScript to create a data structure.

Task: Create a program that manages a simple friend list.

1. Define an object named `people` containing an empty array called `friends`.
2. Create three separate objects, each representing a friend, with properties like `firstName`, `lastName`, and optionally `id`.
3. Add these friend objects to the `friends` array within the `people` object.
4. Output the entire `people` object to the console, displaying your information and your friend list.

Assignment 2:Manipulating an Array: Rearranging Words

Objective:

Rearrange an array using array methods to form the sentence "I am a student of GIAIC".

Steps:

1. **Scrambled Array:**
    - Start with an array of elements in a scrambled order, like:

```
const scrambledArray = ["student", "of",true, 123, "am", "a", "GIAIC", "I"];
```

- **Modify the Array:**

    - Use methods like `split()`, `join()`, `push()`, `pop()`, `shift()`, and `unshift()` to:
        - Convert non-strings (booleans, numbers) to strings if needed.
        - Split elements into character arrays (optional).
        - Rearrange characters or elements in the desired order (modify original array or create temporary arrays).

- **Output the Result:**

    - Use `join()` to combine elements back into a single string: "I am a student of GIAIC".

Assignment 3: Company Product Catalog

Learning Objective: Implement data structures in TypeScript to represent and manage product information.

Task: Create a program to represent a product catalog using an array and perform basic queries.

1. Define an array named `inventory` to store product information.
2. Create three separate objects, each representing a product, with properties like `name`, `model`, `cost`, and `quantity`.
3. Add these product objects to the `inventory` array using an appropriate array method.
4. Access and log the `quantity` property of a specific product (e.g., third product) in the `inventory` array.
5. Explore adding and accessing more elements within the `inventory` array to understand how to manage product data.

Assignment 4: Student List Organizer

Learning Objective: Get comfortable with data structures (objects and arrays) and basic functions in TypeScript.

Tasks:

1. **Student Data:** The provided code defines an interface named `Student` that describes student information like name, senior status (true/false), and whether they've completed their assignments (true/false).
   - Imagine this as a template for organizing student details.
2. **Student List:**
   - An array named `students` stores information about several students using the `Student` template. Think of this array as your class list!
3. **Find Senior Students with Assignments (Optional):**
   - The code (not shown here) has a function that might find students who are seniors and have completed their assignments.
   - Can you guess why this information might be helpful?
4. **Class List Update:**
   - Imagine you need to update your class list! The code (not shown here) might have a function that removes students who haven't completed their assignments (assuming only seniors are responsible).
   - Can you think of any reasons why this might be useful (consider limitations)?