Control Structures

Switch Cases for RPG Game Project

1. **Quest Days of the Week:**

   - Given a variable `dayOfWeek` (type: number), write a switch case that logs a special quest or event happening in your RPG game based on the day of the week (1 for Sunday through 7 for Saturday).

2. **Game Theme Color Setting:**

   - Write a switch case for a variable `themeColor` (type: string) to set the theme color of your RPG game's user interface. Include cases for `'light'` , `'dark'` , `'dungeon'` , and a default case indicating an unsupported color theme.

3. **Quest Notification Status:**

   - Handle multiple cases (`'new'` , `'unread'` , `'unopened'`) in a switch case for a variable `questStatus` (type: string) that logs `"You have new quests awaiting!"` for these cases. Include a default case for `'completed'` quests.

4. **Player Level Classification:**

   - Write a switch case for a variable `playerLevel` (type: string) with cases for levels `'Beginner'` , `'Intermediate'` , `'Advanced'` , `'Expert'` , and `'Master'` . Log a brief description of each level category, and use `break` properly.

5. **Inventory Item Selection:**

   - Write a switch case for a variable `inventoryItem` (type: string) with cases for `'Potion'` , `'Sword'` , `'Shield'` without using `break` . Log the item selected and any subsequent items due to the fall-through behavior of switch cases.

6. **Player Score Ranges:**

- Create a switch case for a variable `playerScore` (type: number) that logs different messages based on score ranges.
    - Define the following score ranges and corresponding messages:
        - 90 or above: "Legend"
        - Between 70 and 89 (inclusive): "Champion"
        - Below 70: "Rookie"
    - Use a switch statement with `playerScore` as the switch expression.
    - Log the appropriate message for each defined score range.
    - Include a default case to handle scores outside the defined ranges, logging a message for players categorized as "Rookie".

7. **Seasonal Quests:**

    - Write a switch case that converts a number (1-12) to the corresponding seasonal quest for a variable `questMonth` (type: number).

        - Define seasonal quests for each month as follows:

            - January (1): "Ice Kingdom Adventure"
            - February (2): "Valentine's Day Special"
            - March (3): "Spring Blossom Hunt"
            - April (4): "Easter Egg Hunt"
            - May (5): "Flower Festival Celebration"
            - June (6): "Summer Solstice Quest"
            - July (7): "Beach Party Bonanza"
            - August (8): "Underwater Expedition"
            - September (9): "Harvest Festival Quest"
            - October (10): "Haunted Mansion Mystery"
            - November (11): "Turkey Trot Adventure"
            - December (12): "Frosty Festivities"

        - Set up a switch statement with `questMonth` as the switch expression.

        - Define each case in the switch statement to correspond to a specific month.
        - Assign the corresponding seasonal quest text to the `questMonth` variable for each case.
        - Include a default case to handle invalid month inputs, logging a message indicating an invalid month for quest.
        - Test the switch case with different values of `questMonth` to ensure it produces the correct seasonal quests.

8. **Player Input Handling:**
    - Write a switch case that checks the type of a variable `playerInput` and logs different messages based on whether the input is a `'string'` , `'number'` , or `'boolean'` .

9. **Game Update Release Schedule:**
    - Write a switch case for a variable `updateMonth` (type: number, 1-12) that logs the game update schedule based on the quarter of the year (`'Q1'` , `'Q2'` , `'Q3'` , `'Q4'`).