PROJECT 2ND: Color Meaning Checker

In this project, you will create a **Color Meaning Checker** application that allows users to input a color name (e.g., "Red", "Blue", "Green") and the program will return the meaning or symbolism of the color. The meanings can be predefined in a dictionary or map. This project will help practice working with maps, strings, and user input in Java.

Small Step-by-Step Guide:

1. **Create a Java Map** (HashMap) to store the color names as keys and their meanings as values (e.g., "Red" -> "Symbolizes passion, energy, and love").
2. **Allow the user to input** the color name using Scanner.
3. Check if the color exists in the **map** and return the corresponding meaning.
4. If the color is not found, prompt the user with a message like "Color not found. Try again."
5. Optionally, allow the user to add new color meanings to the map (this could be done via user input).
6. **Enhance** the application by displaying a list of all available colors if the user asks for it.
7. Implement basic **error handling** for invalid inputs (e.g., blank entries or non-existent colors).

Concepts Used in This Project:

- **Java Collections**: HashMap, Scanner
- **String Manipulation** for input and comparison
- **User Input Handling**
- **Error Handling** (e.g., invalid color names)
- **Working with Maps** for key-value data storage

Why We Have Chosen This Project:

- Helps learn **Java Maps** (key-value storage) and **basic user input handling**.
- Great for practicing **string comparison** and **validation**.
- Can be easily expanded to include more features (e.g., adding colors dynamically).
- Introduces the concept of **error handling** in a simple, interactive way.