**Programming Problem: Getting Ready**

**Problem:**

You are in your house wearing pajamas. You must be appropriately dressed for the temperature before leaving your house.

**Challenge:**

Your challenge is to programmatically process a list of commands for getting ready, enforce related rules, and display appropriate output. The implemented rules should be flexible to change. The implementation of the described problem should follow SOLID principles of object oriented design.

**Inputs:**

1. Temperature type (one of the following)
   * HOT
   * COLD
2. Comma separated list of numeric commands

|  |  |  |  |
| --- | --- | --- | --- |
| Command | Description | Hot Response | Cold Response |
| 1 | Put on footwear | “sandals” | “boots” |
| 2 | Put on shirt | “t-shirt” | “shirt” |
| 3 | Put on jacket | fail | “jacket” |
| 4 | Put on pants | “shorts” | “pants” |
| 5 | Leave house | “leaving house” | “leaving house” |
| 6 | Take off pajamas | “Removing PJs” | “Removing PJs” |

**Assumptions:**

* Initial state is in your house with your pajamas on

**Rules:**

* Pajamas must be taken off before anything else can be put on
* Only 1 piece of each type of clothing may be put on
* You cannot put on a jacket when it is hot
* Pants must be put on before shoes
* The shirt must be put on before jacket
* You cannot leave the house until all items of clothing are on (except a jacket when it’s hot)
* If an invalid command is issued, respond with “fail” and stop processing commands

**Examples**

* Input: HOT 6, 4, 2, 1, 5 Output: Removing PJs, shorts, t-shirt, sandals, leaving house
* Input: COLD 6, 4, 2, 3, 1, 5 Output: Removing PJs, pants, shirt, jacket, boots, leaving house
* Input: COLD 4 Output:  fail
* Input: HOT 6, 4, 3 Output:  Removing PJs, shorts, fail
* Input: HOT 6, 4, 4 Output:  Removing PJs, shorts, fail
* Input: COLD 6, 4, 2, 3, 5 Output:  Removing PJs, pants, shirt, jacket, fail

**Directions**

Please submit your solution in .NET 4.0 using C# and provide all source, test and build support files. The project structure is up to you, assume that this code will be deployed to production and your peers will be maintaining the code going forward.

**Criteria**

You will primarily be judged on the code directly related to the implementation of the stated problem and business rules:

* Use of Object Oriented Principles (SOLID)
* Code Legibility
* Ease of Maintenance
* Use of recognizable best practices and patterns
* Submission of a compiling and fully working solution
* Correct implementation of rules

Secondary evaluation criteria will be the usage and evident knowledge of the frameworks and methodologies specified in the job description.

We value creativity and initiative to learn new technology; however, be advised that candidates that focus solely on the primary criteria will be more successful than candidates that focus instead on intricate UI and usage of a breadth of technologies.