PROJECT DOCUMENATION

Instructions:

1. **Define the ListManager Class:**
   - The class should manage a list of items using a std::vector<std::string>.
   - Implement methods for:
     - Adding an item to the list.
     - Removing an item from the list by its index.
     - Removing an item from the list by its value.
1. **Add Item Function:**
   - Define the addItem method to add a string item to the items vector.
   - Provide feedback to the user that the item has been added.

2

Instructions:

- **Remove Item by Index Function:**
  - Define the removeItemByIndex method to remove an item based on its index in the items vector.
  - Check if the index is within the valid range.
  - Provide feedback if the item is removed or if the index is out of range.
- **Remove Item by Value Function:**
  - Define the removeItemByValue method to remove an item based on its value.
  - Use std::find to locate the item in the vector.
  - Provide feedback if the item is removed or if the item is not found.
-

Instructions:

3. **Display Items Function:**
   - Define the displayItems method to display all items in the list.
   - Provide a message if the list is empty.
   - User Interface Menu:
   - Create a showMenu function to display the menu options to the user.
   - **Main Function:**
   - **I**mplement the main loop to display the menu and process user choices.
   - Handle user input for menu choices and call the appropriate.
   - Validate user input to handle invalid choices and ensure the program does not crash.

Challenges Faced and Their Solutions

- Include Necessary Headers:
- Ensure you have included the correct headers for the functions and types you are using. Specifically, you need to include headers for algorithms (like std::find) and numeric limits (like std::numeric_limits).
- Correct Iterator Declaration:
- When using std::find to search for an item in a list, make sure you correctly declare the iterator. This iterator is used to check if the item exists and to remove it if found.
- Handle std::numeric_limits:
- When clearing invalid input from the stream, use std::numeric_limits correctly to ignore the maximum number of characters. This ensures your program handles invalid inputs gracefully.

Challenges Faced and Their Solutions

- Implement the functions to add items, remove items by index, remove items by value, and display items. Ensure these functions handle edge cases, such as when the list is empty or when an invalid index is provided.
- Input Validation:
- In your main loop, handle invalid inputs gracefully. If the user enters non-numeric input when a number is expected, clear the error flag and ignore the invalid input to avoid crashing or getting stuck in an infinite loop.
- Compile and Test:
- After making the necessary changes, compile your program.
- Test all functionalities thoroughly, including adding items, removing items by index, removing items by value, and displaying items.
- Specifically test edge cases, such as removing items from an empty list or providing an invalid index.

LinkedIn Post Link:
https://www.linkedin.com/feed/update/urn:li:activity:7214011679758553090/

GitHub Post Link:
https://github.com/Arubakhan 22/Handling-Data-Types-and-Variables