

Shopping List Command Line Application

“Create a Python command line program that manages a persistent shopping list. The program must provide a simple interactive menu, allow adding, removing, showing, and saving items, and persist the list to a text file so items remain between runs.”

Overview

Create a Python command line program that manages a persistent shopping list. The program must provide a simple interactive menu, allow adding, removing, showing, and saving items, and persist the list to a text file so items remain between runs.

Learning Objectives

- Implement a Python class to encapsulate program state and behavior.
- Read from and write to files for simple persistence.
- Build a command loop that handles user input and common edge cases.
- Apply basic input validation and provide clear user feedback.

Requirements

- Language: Python 3.x
- Persistence file: shopping_list.txt in the program directory.
- Class name: ShoppingList
- Required public methods:
 - - `__init__` — initialize state and load items from file.
 - - `start_menu` — print the menu/instructions.
 - - `add_to_list(item)` — add a single item (case normalized) if not already present.
 - - `remove_item(item)` — remove a single item if present.
 - - `show_shopping_list()` — print the current list.
 - - `save_items()` — write the list to the file.
 - - `load_items()` — read items from the file into memory.
- Command loop behavior:
 - - Prompt the user with '`>`' and accept input until the user issues the DONE command or an EOF (Ctrl+D / Ctrl+Z) occurs.

- - Commands are case insensitive. Support at least these commands: DONE, HELP, SHOW, REMOVE, or any item to add.
- Item normalization: Capitalize items (first letter uppercase, rest lowercase) before storing and comparing.
- User feedback: Confirm actions and show current count.
- Error handling: Gracefully handle file I/O errors and EOFError from input() by saving and exiting with a friendly message.

Sample Session

What do you want to add to your shopping list?

Enter 'DONE' to stop adding items.

Enter 'HELP' for additional info.

Enter 'SHOW' to see your shopping list.

Enter 'REMOVE' to remove an item from your shopping list.

> milk

Milk was added to your shopping list.

You have 1 items on your list.

> eggs

Eggs was added to your shopping list.

You have 2 items on your list.

> show

My Shopping List:

- Milk

- Eggs

> remove

My Shopping List:

- Milk

- Eggs

What do you want to remove?: eggs

Eggs was removed from your shopping list.

You have 1 items on your list.

> done

See you soon!

Your shopping list has been saved.

My Shopping List:

- Milk

Constraints and Grading Criteria

Criterion — Points — Notes

Program runs without syntax errors — 10 — Must execute under Python 3.x

Class structure and required methods — 20 — `ShoppingList` with listed methods

File persistence implemented correctly — 25 — Items persist between runs

Correct command loop and input handling — 20 — Commands and EOF handling

Proper item normalization and duplicate prevention — 10 — Capitalization and uniqueness

User feedback and error handling — 10 — Clear messages and graceful failures

Test Cases

- Empty start: If shopping_list.txt does not exist, program starts with an empty list and prints no items on SHOW.
- Add duplicate: Add `apple`, then `Apple` — second add should report the item already exists.
- Remove missing: Attempt to remove `banana` when not present — program should report not found.
- Persistence: Add items, exit with DONE, restart program — previously added items should be loaded.
- EOF handling: Send EOF at the prompt — program should save and exit cleanly.

Submission Instructions

- Submit a single Python file named shopping_list.py.
- The file must be runnable as a script (include `if __name__ == '__main__': main()`).
- Include brief comments at the top with your name and a one-line description.
- Optionally include a short README describing any extra features you implemented.

Optional Extensions for Extra Credit

- Allow items to include quantities (e.g., `Milk x2`) and parse/store quantity separately.
- Add a `CLEAR` command to empty the list after confirmation.
- Implement alphabetical sorting when showing the list.
- Provide a search command to find items by substring.

End of problem statement.