

Tameeka Marie Hannon

CS 499

Milestone 3 Narrative

Narrative: Enhancements to the Inventory Management System

Artifact Description

The artifact for this enhancement is an inventory management system implemented in Python.

The original code provided a basic structure to manage inventory items, including their names, prices, and quantities. It allowed for the display of the inventory and the addition of new items.

However, the system lacked the functionality to handle fractional prices (i.e., prices less than \$1), and it did not support sorting inventory by item name, price, or quantity. Additionally, the inventory was not persistent across program runs—it was only available during the current session. This enhancement improves the system by adding these missing functionalities.

Justification for Inclusion

This inventory management system was selected as the artifact for my ePortfolio because it showcases my ability to implement fundamental concepts in data structures, algorithms, and file management. The system makes use of lists (a fundamental data structure in Python) to store inventory items, and it provides operations for modifying and querying the inventory, such as adding items and sorting them. The enhancement focuses on solving two key issues:

1. **Handling Fractional Prices:** The original version could only accept whole numbers for prices. This was addressed by modifying the price input to accept decimal values, allowing for more realistic pricing for items that cost less than \$1.

2. **Sorting Inventory:** The ability to sort inventory by item name, price, or quantity is a valuable feature for an inventory management system. I added sorting functionality that allows the user to choose a field (name, price, or quantity) to sort the inventory in ascending order.

Additionally, I integrated file handling so that the inventory is saved to a `inventory.json` file and is loaded when the program starts, providing persistence across different program runs. This ensures that the user does not lose their data when the program exits, making the system more robust and practical for real-world use.

Skills and Abilities in Algorithms and Data Structures

The enhancement primarily demonstrates my understanding of:

- **Data Structures:** The system uses lists to manage the inventory. Each item in the inventory is represented as a dictionary containing its name, price, and quantity. This structure is simple but effective for the purpose of this assignment.
- **Algorithms:** The sorting functionality leverages Python's built-in sorting capabilities, which allow the inventory to be sorted by any of the item attributes. This enhances the system's usability, as users can now sort by name, price, or quantity with a single command. The sorting is done using a key function in the `sort()` method, which allows for sorting based on different attributes.
- **File Handling:** The persistence of the inventory data is ensured by using the `json` module. The inventory is saved to a file after every update, and it is reloaded when the program starts. This ensures that the inventory data is retained between sessions, making the system more practical for real-world use cases.

Enhancements and Improvements

The inventory management system was significantly improved by:

- Adding the ability to handle fractional prices (items costing less than \$1).
- Introducing a sorting feature, allowing the user to sort the inventory by name, price, or quantity.
- Implementing file handling using JSON, so the inventory data is saved between sessions and can be loaded when the program starts. These improvements address the limitations of the initial version and enhance the overall functionality of the system, making it more user-friendly and practical for managing inventory.

Meeting Course Outcomes

The planned enhancements met the course outcomes I set for myself in Module One:

- **Data Structures and Algorithms:** I effectively used lists to store inventory and implemented a sorting algorithm to organize the inventory based on user input. This demonstrates my ability to design and implement computing solutions that utilize algorithmic principles.
- **Persistence and File Management:** By adding the ability to load and save inventory data using JSON, I improved the program's ability to manage data over time. This aligns with the course outcome of designing and implementing computing solutions that deliver value and can be applied to real-world problems.

Reflection on the Enhancement Process

During the enhancement process, I learned several important lessons:

- I encountered challenges when modifying the program to handle fractional prices, particularly in how to validate the user input to ensure that it was a valid number. I used exception handling to catch errors and prompt the user to enter valid input, which made the program more robust.
- Implementing the sorting functionality required me to think about the user experience—how would users want to sort the inventory, and what would be the most intuitive way to implement this? I decided to allow sorting by any of the three attributes (name, price, and quantity) to give users maximum flexibility.
- Adding file handling was a key challenge because I needed to ensure that the inventory was saved and loaded correctly. I learned how to use Python's json module to read and write to files, which was an essential skill for making the system more useful over time.

Overall, this enhancement process improved my understanding of algorithms, data structures, and file management. The system is now more user-friendly and better suited for managing real-world inventory data. I also gained a deeper understanding of how to build and enhance Python programs that solve practical problems.