Practice Exam: Wizard Hat Shop – Order Queue

Overview

You will be writing a program to help manage hat orders at a wizard's shop. Each order will include the customer's name, the size of the hat, and the type of enchantment placed on it.

Your program should allow the user to add orders, fulfill them in the order they were received, view all pending orders, count how many hats are waiting for a specific enchantment, and calculate the average hat size across all pending orders.

Functions Required:

You must write two functions for this practice exam:

- *count_enchantment:*
  - Takes in the list of orders and an enchantment name (string). Returns how many orders in the list use that enchantment. Ignore capitalization and extra spaces when checking the enchantment.
- *average_hat_size:*
  - Takes in the list of orders. Returns the average hat size (float) across all pending orders. If there are no orders, return 0.0.

Logical Flow

Start your program with an empty list. You'll eventually fill this list with wizard hat orders. Each order will be stored as a dictionary with three keys: 'customer', 'size', and 'enchantment'. (In other words, you'll eventually have a list with several dictionaries inside. At the start it will just be an empty list though)

Your program should then enter a loop where it repeatedly displays a menu like this:

```
Wizard Hat Shop – Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit

Enter an option (1–6):
```

After completing the action for the chosen option, the menu should be displayed again until the user selects Exit.

Option 1: Add an order

- Ask the user for a customer name.
- Ask the user for a hat size in inches (float).
- Ask the user for an enchantment (like invisibility, luck, wisdom, etc) (string).
- Store this information in a dictionary and add it to the end of the orders list.

Option 2: Fulfill next order

- If there are no orders in the list, print a message that says so.

- Otherwise, remove the first order from the list and display its information (customer, size, enchantment). And print out a message saying this order has been fulfilled.

Option 3: List all pending orders
- If there are no pending orders, print a message that says so.
- Otherwise, print each order in the list with its index (display the index #, but start at 1 instead of 0), customer name, size, and enchantment.

Option 4: Count orders by enchantment
- Ask the user to enter an enchantment (string).
- Call your *count_enchantment* function (using the entered string and the list of orders as arguments. It should return an integer) to count how many orders are waiting with that enchantment.
    - For an extra challenge, make it so it correctly counts even if the enchantment is capitalized differently in the list of orders.
- Print the result.

Option 5: Show average hat size
- Call your *average_hat_size* function using the orders list as an argument. It should return a float that is the calculated average hat size.
- Print the result, formatted to two decimal places.

Option 6: Exit
- Print a goodbye message and end the program.

Any other input
- If the user enters anything other than 1–6, print: Invalid choice, try again!

Rubric:
Since this is just a practice question, there isn't a detailed rubric. You can check your answer against the solution problem, or you can ask the TAs or professor if you want feedback on your practice problem.

Example Output

```
Wizard Hat Shop – Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit

Enter an option (1-6): 1

Customer name: Merlin
Hat size (e.g., 7.25): 7.5
Enchantment (e.g., invisibility, luck, wisdom): Invisibility
Added order for Merlin: size 7.5, enchantment 'invisibility'.
```

```
Wizard Hat Shop — Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit

Enter an option (1-6): 1

Customer name: Morgana
Hat size (e.g., 6.75): 6.75
Enchantment (e.g., invisibility, luck, wisdom): Luck
Added order for Morgana: size 6.75, enchantment 'luck'.

Wizard Hat Shop — Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit

Enter an option (1-6): 3

Pending Orders:
1. Merlin | size 7.5 | invisibility
2. Morgana | size 6.75 | luck

Wizard Hat Shop — Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit

Enter an option (1-6): 2

Fulfilling order:
  Customer: Merlin
  Size: 7.5
  Enchantment: invisibility

Wizard Hat Shop — Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit
```

```
Enter an option (1-6): 4

Enter an enchantment to count: luck
There are 1 orders with enchantment 'luck'.

Wizard Hat Shop — Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit

Enter an option (1-6): 5

Average hat size across pending orders: 6.75

Wizard Hat Shop — Order Queue:

1. Add an order
2. Fulfill next order
3. List all pending orders
4. Count orders by enchantment
5. Show average hat size
6. Exit

Enter an option (1-6): 6

Closing the order book. May your hats fit and your charms never fizzle!
```