# 2CST8130 – Data Structures

# Assignment 1 – Test Plan

# Written By - Ryan Nguyen: 041137485

## Main menu exception handling

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| * 1. Good - User inputs valid option | 1 | Display next menu | Do you wish to add a fruit(f), vegetable(v), preserve(p), or juice(j)? | Pass |
| * 1. Bad - User inputs integer that is not a listed option | 9 | Error message, redisplay main menu | Incorrect value entered | Pass |
| * 1. Bad - User inputs non-integer value | e e | Error message, redisplay main menu | Incorrect value entered | Pass |

## Option 2: Display current inventory

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| 1. Good case - Display an empty inventory | 2 | Header with no item records | Inventory: | Pass |
| 1. Good case - Display an inventory with items | 2 | Display the record(s) | Inventory:  Item: 111 Granny Smith Apple 234 price: $0.25 cost: $0.10 orchard supplier: Niagara orchard | Pass |

## Option 1: add item to inventory

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| 1. Good case - User chooses option 1 from main menu | 1 | Display prompt to choose item type | Do you wish to add a fruit(f), vegetable(v), preserve(p), or juice(j)? | Pass |
| 1. Bad case - User enters invalid item type | e | Error message, display prompt again | Invalid entry | Pass |
| 1. Good case - User enters valid item type | f | Next Prompt | Enter the code for the item: | Pass |
| 1. Bad case - User enters not-integer item code | a | Error message; redisplay prompt | Invalid code  Enter the code for the item: | Pass |
| 1. Bad case - User enters item code that already exists | 123 | Display error message. Redisplay prompt | Item code already exists  Enter the code for the item: | Pass |
| 1. Good case - User enters valid item code | 123 | Next prompt | Enter the name for the item: | Pass |
| 1. Good case - User enters anything for item name | apple | Next prompt | Enter the quantity for the item: | Pass |
| 1. Bad case - User enters 0 for item quantity | 0 | Error message, redisplay prompt | Invalid entry  Enter the quantity for the item: | Pass |
| 1. Bad case - User enters negative number for item quantity | -1 | Error message, redisplay prompt | Invalid entry  Enter the quantity for the item: | Pass |
| 1. Good case - User enters valid item quantity | 10 | Next prompt | Enter the cost of the item: | Pass |
| 1. Bad case - User enters 0 for item cost | 0 | Error message, redisplay prompt | Invalid entry  Enter the cost of the item: | Pass |
| 1. Bad case - User enters negative item cost | -1 | Error message, redisplay prompt | Invalid entry  Enter the cost of the item: | Pass |
| 1. Good case -User enters valid item cost (>= 0) | 2 | Display next prompt | Enter the sales price of the item: | Pass |
| 1. Bad case - User enters 0 for sales price | 0 | Error message, redisplay prompt | Invalid entry  Enter the sales price of the item: | Pass |
| 1. Bad case - User enters negative sales price | -1 | Error message, redisplay prompt | Invalid entry  Enter the sales price of the item: | Pass |
| 1. Good case - User enters valid sales price | 3 | Display next prompt | Enter the name of the orchard supplier: | Pass |
| 1. Bad case - User attemps to add item to inventory when the inventory is full. | 1 | Display error messaage | Inventory at max capacity | Pass |

## Option 3: buy item(s)

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| 1. Bad case - User enters a code that does not exist in inventory | 999 | Display error message. Display main menu | Code not found in inventory...  Error...could not buy item | Pass |
| 1. Bad case – User tries to buy when there are no items in inventory | 3 | Display error message. Display main menu | Error...could not buy item | Pass |
| 1. Bad case – User enters non-int item code | e e | Display error message. Redisplay prompt | Invalid code  Enter the code for the item: | Pass |
| 1. Good case - User enters item code that exists in inventory | 100 | Display next prompt | Enter valid quantity to buy: | Pass |
| 1. Good case -User enters positive, non-zero item quantity | 10 | Return to main menu. Quantity is updated | <main menu> | Pass |
| 1. Bad case - User enters 0 for quantity to buy | 0 | Display error message. Return to main menu | Invalid quantity...  Error...could not buy item | Pass |
| 1. Bad case - User enters negative quantity to buy | -1 | Display error message. Return to main menu | Invalid quantity...  Error...could not buy item | Pass |
| 1. Bad case - User attempts to buy items when the inventory is empty | 3 | Display error message. Return to main menu | Error...could not buy item | Pass |

## Option 4: sell item(s)

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| 1. Good case - User enters item code that exists | 100 | Display next prompt | Enter valid quantity to sell: | Pass |
| 1. Bad case - User enters item code that does not exist in the inventory | 999 | Code not found in inventory...  Error...could not sell item | Code not found in inventory...  Error...could not sell item | Pass |
| 1. Bad case – user attempts to sell items when there are no records in inventory | 4 | Display error message. Return to main menu | Error...could not sell item | Pass |
| 1. Good case - User enters valid quantity | 10 | Return to main menu. The item’s quantity is updated. | <main menu> | Pass |
| 1. User enters 0 for item quantity | 0 | Error message. Return to main menu | Invalid quantity...  Error...could not sell item | Pass |
| 1. User enters quantity greater than amount in storage | 100 | Insufficient stock in inventory... | Insufficient stock in inventory... | Pass |
| 1. User attempts to sell items without a record in inventory | 4 | Error...could not sell item | Error...could not sell item | Pass |

## Option 5: search for item by code

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| 1. Bad case - User searches for code that does not exist in inventory | 999 | Code not found message | Code not found in inventory... | Pass |
| 1. Good case – User searches for code that does exist in inventory | 322 | Display the items’s data | Item: 322 Vine Ripe Tomato 100 price: $0.35 cost: $0.10 farm suppllier: McDonald's Farm | Pass |

## Option 6: Save inventory to file

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| 1. Good case - User chooses option 7 | 6 | Display prompt for file name to output to. | Enter the filename to save to: | Pass |
| 1. Good case - User enters a file name | MyStockData.txt | Return to main menu. New file is created with the inventory data | <main menu> | Pass |

## Option 7: Read Inventory from File

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| 1. Good case – user chooses option 7 | 7 | Display prompt for file name to read from. | Enter the filename to read from: | Pass |
| 1. Good case - User enters a file name that exists | MyStockData2.txt | Return to main menu. Inventory is updated with the items from the file | <main menu> | Pass |
| 1. Bad case - User enters a file name that does not exist | e.txt | Error message, return to main menu | File Not Found, ignoring... | Pass |
| 1. Bad case - User reads from a file but the file contains item codes that already exist in the inventory | MyStockData2.txt | When the duplicate item code is read, stop reading; display error message; return to main menu | Item code already exists  Error Encountered while reading the file, aborting... | Pass |

## Option 8: Exit

| Description of Test | Input Values | Expected Output | Actual Output / Result | Pass/Fail |
| --- | --- | --- | --- | --- |
| a. User chooses option 8 | 8 | Display exiting message | Exiting... | Pass |

Its expected that you:

<add more tables if you have more features to test>