# CST8130 – Data Structures

# Assignment <1> – Test Plan

# Written By - <Yuliia Topalova>

## < Main Menu>

| Description of Test | Input Values | Output / Result | Pass/Fail |
| --- | --- | --- | --- |
| * 1. Add Item to Inventory | 1 | Open menu with add info | Pass |
| * 1. Display Current Inventory | 2 | Displays Items in Inventory | Pass |
| * 1. Buy Item(s) | 3 | Buying menu is appeared | Pass |
| * 1. Sell Item(s) | 4 | Selling menu is appeared | Pass |
| * 1. To Exit | 5 | Exit program with "Exiting..." | Pass |
| * 1. Any other number: | 6,7,8,9 | Outputs error “Something went wrong...” and asks to renter the value. | Fail |
| * 1. Any letter | A,B,C,D | Outputs error “Something went wrong...” and asks to renter the value. | Fail |

## <Add Item to Inventory Menu>

| Description of Test | Input Values | Output / Result | Pass/Fail |
| --- | --- | --- | --- |
| 1. If F selected | F | Open menu for add new fruit | Pass |
| 1. If V selected | V | Open menu for add new vegetable | Pass |
| 1. If P selected | P | Open menu for add new preserve | Pass |
| 1. Any letter | A,B,C,D | Outputs error “Something went wrong...” and asks to renter the value. | Fail |
| 1. If item code exists | int value | Outputs a message that code already exists | Pass |

## <Add new Fruit item>

| Description of Test | Input Values | Output / Result | Pass/Fail |
| --- | --- | --- | --- |
| 1. Enter the code for the item | Int value | Saves the code and outputs the name line | Pass |
| 1. Enter the name for the item | Any input | Saves the name and outputs the quantity line | Pass |
| 1. Enter the quantity for the item | Positive int | Saves the quantity and outputs the cost line | Pass |
| 1. Enter the quantity for the item | Negative int | Outputs Invalid input | Fail |
| 1. Enter the cost of the item | Positive number | Saves the cost and outputs the price line | Pass |
| 1. Enter the cost of the item | Negative number | Outputs Invalid input | Fail |
| 1. Enter the sales price of the item | Positive number | Saves the price and outputs the orchard supplier line | Pass |
| 1. Enter the sales price of the item | Negative number | Outputs Invalid input | Fail |
| 1. Enter the name of the orchard supplier | Any input | Save the name | Pass |

## <Add new Vegetable item>

| Description of Test | Input Values | Output / Result | Pass/Fail |
| --- | --- | --- | --- |
| 1. Enter the code for the item | Int value | Saves the code and outputs the name line | Pass |
| 1. Enter the name for the item | Any input | Saves the name and outputs the quantity line | Pass |
| 1. Enter the quantity for the item | Positive int | Saves the quantity and outputs the cost line | Pass |
| 1. Enter the quantity for the item | Negative int | Outputs Invalid input | Fail |
| 1. Enter the cost of the item | Positive number | Saves the cost and outputs the price line | Pass |
| 1. Enter the cost of the item | Negative number | Outputs Invalid input | Fail |
| 1. Enter the sales price of the item | Positive number | Saves the price and outputs the orchard supplier line | Pass |
| 1. Enter the sales price of the item | Negative number | Outputs Invalid input | Fail |
| 1. Enter the name of the farm supplier | Any input | Save the name | Pass |

## <Add new preserve item>

| Description of Test | Input Values | Output / Result | Pass/Fail |
| --- | --- | --- | --- |
| 1. Enter the code for the item | Int value | Saves the code and outputs the name line | Pass |
| 1. Enter the name for the item | Any input | Saves the name and outputs the quantity line | Pass |
| 1. Enter the quantity for the item | Positive int | Saves the quantity and outputs the cost line | Pass |
| 1. Enter the quantity for the item | Negative int | Outputs Invalid input | Fail |
| 1. Enter the cost of the item | Positive number | Saves the cost and outputs the price line | Pass |
| 1. Enter the cost of the item | Negative number | Outputs Invalid input | Fail |
| 1. Enter the sales price of the item | Positive number | Saves the price and outputs the orchard supplier line | Pass |
| 1. Enter the sales price of the item | Negative number | Outputs Invalid input | Fail |
| 1. Enter the size of the jar in millilitres | Positive int | Save the name | Pass |
| 1. Enter the size of the jar in millilitres | Negative int | Invalid input | Fail |

## <Buy Item>

| Description of Test | Input Values | Output / Result | Pass/Fail |
| --- | --- | --- | --- |
| 1. When code doesn't exist | int value | Error...could not buy item | Fail |
| 1. When code is found | int value | Accepts entry and prompts for quantity | Pass |
| 1. If quantity is valid | int value | Updates quantity of stock for item code | Pass |
| 1. If quantity is not valid | int value | Error...could not buy item | Fail |

## <Sell Item>

| Description of Test | Input Values | Output / Result | Pass/Fail |
| --- | --- | --- | --- |
| 1. When code doesn't exist | int value | Error...could not buy item | Fail |
| 1. When code is found | int value | Accepts entry and prompts for quantity | Pass |
| 1. If quantity is valid | int value | Updates quantity of stock for item code | Pass |
| 1. If quantity is not valid | int value | Error...could not buy item | Fail |