

Documentation Template  
CSC 414 – Software Design  
James Setliff  
November 15, 2020

## 1.0 – Scope

### 1.1 – Identification

Point of Sale/Inventory Management System v 0.0.0.0.1

### 1.2 – System Overview

POSIMS requires Microsoft Windows 10 and Python 3.9.0. Some functionalities will work on any operating system, given the nature of the Python interpreter, but file writing and input/output methods cannot be guaranteed on other systems. The software will provide basic calculation methods for billing, ordering, and inventory management. POSIMS is designed to be light-weight and easily deployable on any hardware that meets the minimum requirements previously mentioned.

### 1.3 – Document Overview

This document serves as a point of reference for the design, implementation, testing, and maintenance of the software.

## 2.0 – References

Python 3.9.0 documentation - <https://docs.python.org/3/>

How to use a keyboard - <https://www.wikihow.com/Use-a-Computer-Keyboard>

### 3.0 – Requirements

Requirement #	Description
<b>R1.0</b>	Program shall ensure a running inventory is stored on disk and maintained through multiple use cycles.
<b>R1.1</b>	Program shall facilitate adding to and removing from the inventory.
<b>R1.2</b>	Program shall facilitate viewing the inventory from inside the program.
<b>R1.3</b>	Program shall, at a minimum, track item name and quantity.
<b>R1.4</b>	If an item is sold that is not currently in inventory, the program shall add it to build a more accurate record over time.
<b>R1.5</b>	Program shall support user requested exports of the inventory.
<b>R1.6</b>	Program shall support a basic, ledger-style transaction system and write transactions to disk.
<b>R1.7</b>	Program shall display an editable message for notifying employees of important information.

#### 4.0 – Design

<b>Design #</b>	<b>Ref Req#</b>	<b>Design Description</b>
<b>D1</b>	R1.0	Write functions to read from file on disk at program launch and write to file on disk at program exit.
<b>D2</b>	R1.3	Use dictionary data structure to store working inventory while program is running. Item name will suffice as unique key, and quantity can be held in the value. This should meet minimum requirements and can be expanded to use a different key (i.e. serial number) and hold additional values if necessary, in future.
<b>D3</b>	R1.1	Write function to manipulate inventory during runtime. A standalone function is necessary to adjust these values for inventory intake or when a transaction has not occurred.
<b>D4</b>	R1.2	Write function to iterate through dictionary items and display them in a sensible format for the end user.
<b>D5</b>	R1.5	Write function for user to request export, perhaps reuse the write function from D1 to write the new file. New file needs to have a unique name to prevent accidental overwrites.
<b>D6</b>	R1.6 R1.4	Write function to facilitate sales – must capture basic information (customer, item, quantity, total price for starters?) and write to ledger. Adjust inventory at the same time the transaction is processed.
<b>D7</b>	R1.7	Add “message of the day” to main screen. Write function to adjust message.

## 5.0 – Test Plan

As this program serves as an entry point for a new over-arching system, testing on this will be limited to positive path and end-user functional testing. Integration testing is presumed to be of little or no consequence as there is no system in place to integrate with, and likewise, regression testing is impossible to conduct with no previous version in place. With that being stated, testing responsibilities should fall on knowledgeable, experienced users who understand the requirements listed above and insight into inventory management.

### 5.1 – Test Procedures

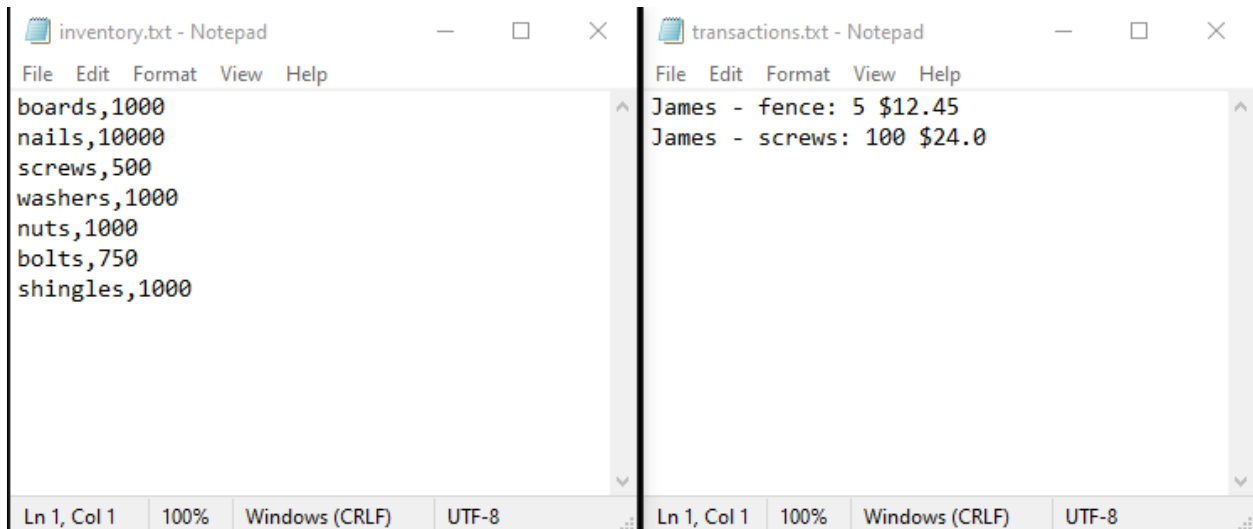
- 1.) Ensure software is deployed on systems that match prerequisites stated in this document. Contact your IT personnel for assistance.
- 2.) Ensure all Project Requirements are met in accordance with the project requirements table (reproduced below):

Requirement #	Description
<b>R1.0</b>	Program shall ensure a running inventory is stored on disk and maintained through multiple use cycles.
<b>R1.1</b>	Program shall facilitate adding to and removing from the inventory.
<b>R1.2</b>	Program shall facilitate viewing the inventory from inside the program.
<b>R1.3</b>	Program shall, at a minimum, track item name and quantity.
<b>R1.4</b>	If an item is sold that is not currently in inventory, the program shall add it to build a more accurate record over time.
<b>R1.5</b>	Program shall support user requested exports of the inventory.
<b>R1.6</b>	Program shall support a basic, ledger-style transaction system and write transactions to disk.
<b>R1.7</b>	Program shall display an editable message for notifying employees of important information.

- 3.) Record Pass/Fail for each testing requirement, along with any supporting information that may help identify and address your issues in the future. Additionally, all feedback is welcome and will be taken into consideration when designing future improvements to this software. Please help us to help you and fill in as much as possible!

## APPENDIX A – Test Results

### R1.0 & R1.3 - Starting inventory and transaction log files, item names & quantities



The screenshot shows two Notepad windows side-by-side. The left window, titled 'inventory.txt - Notepad', contains a list of items and their quantities: boards,1000; nails,10000; screws,500; washers,1000; nuts,1000; bolts,750; and shingles,1000. The right window, titled 'transactions.txt - Notepad', contains two transaction entries: James - fence: 5 \$12.45 and James - screws: 100 \$24.0. Both windows have a status bar at the bottom indicating 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.

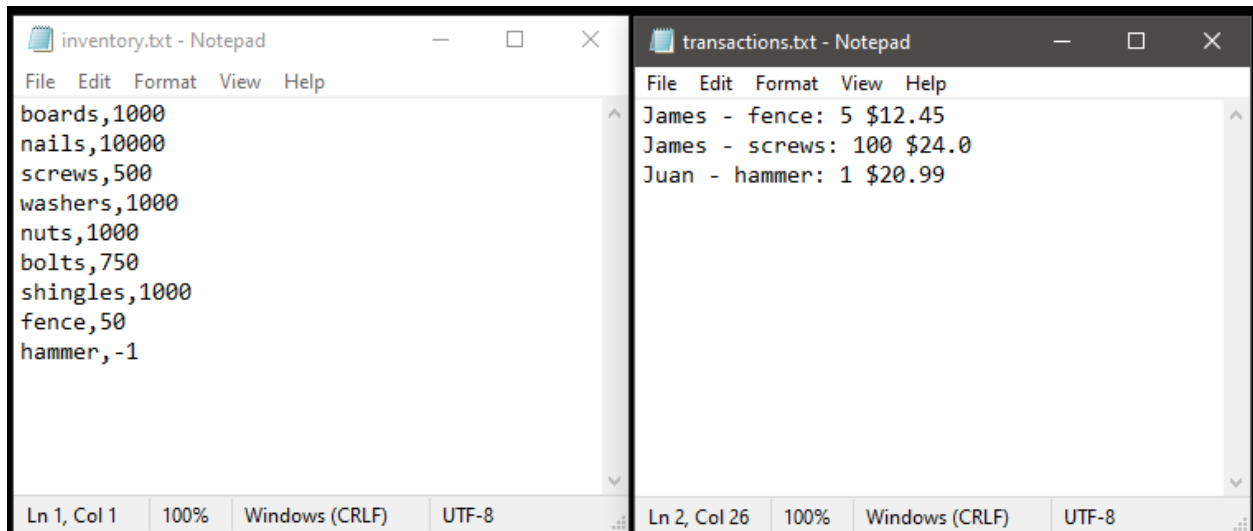
```
inventory.txt - Notepad
File Edit Format View Help
boards,1000
nails,10000
screws,500
washers,1000
nuts,1000
bolts,750
shingles,1000

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

transactions.txt - Notepad
File Edit Format View Help
James - fence: 5 $12.45
James - screws: 100 $24.0

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

### Resulting inventory and transaction log files



The screenshot shows the same two Notepad windows after transactions. The left window, 'inventory.txt - Notepad', now includes 'fence,50' and 'hammer,-1' at the bottom of the list. The right window, 'transactions.txt - Notepad', has a third entry: Juan - hammer: 1 \$20.99. The status bar for the right window now indicates 'Ln 2, Col 26'.

```
inventory.txt - Notepad
File Edit Format View Help
boards,1000
nails,10000
screws,500
washers,1000
nuts,1000
bolts,750
shingles,1000
fence,50
hammer,-1

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

transactions.txt - Notepad
File Edit Format View Help
James - fence: 5 $12.45
James - screws: 100 $24.0
Juan - hammer: 1 $20.99

Ln 2, Col 26 | 100% | Windows (CRLF) | UTF-8
```

R1.1 & R1.2 & R1.3 – Adding items to inventory and viewing inventory, item names and quantities

```
C:\Windows\py.exe
Inventory Menu

Please select from the following menu items...

1.) View Inventory
2.) Add Item to Inventory
3.) Export Inventory
4.) Return to Main Menu
#> 2
Enter new item name: fence
Enter new quantity: 50
```

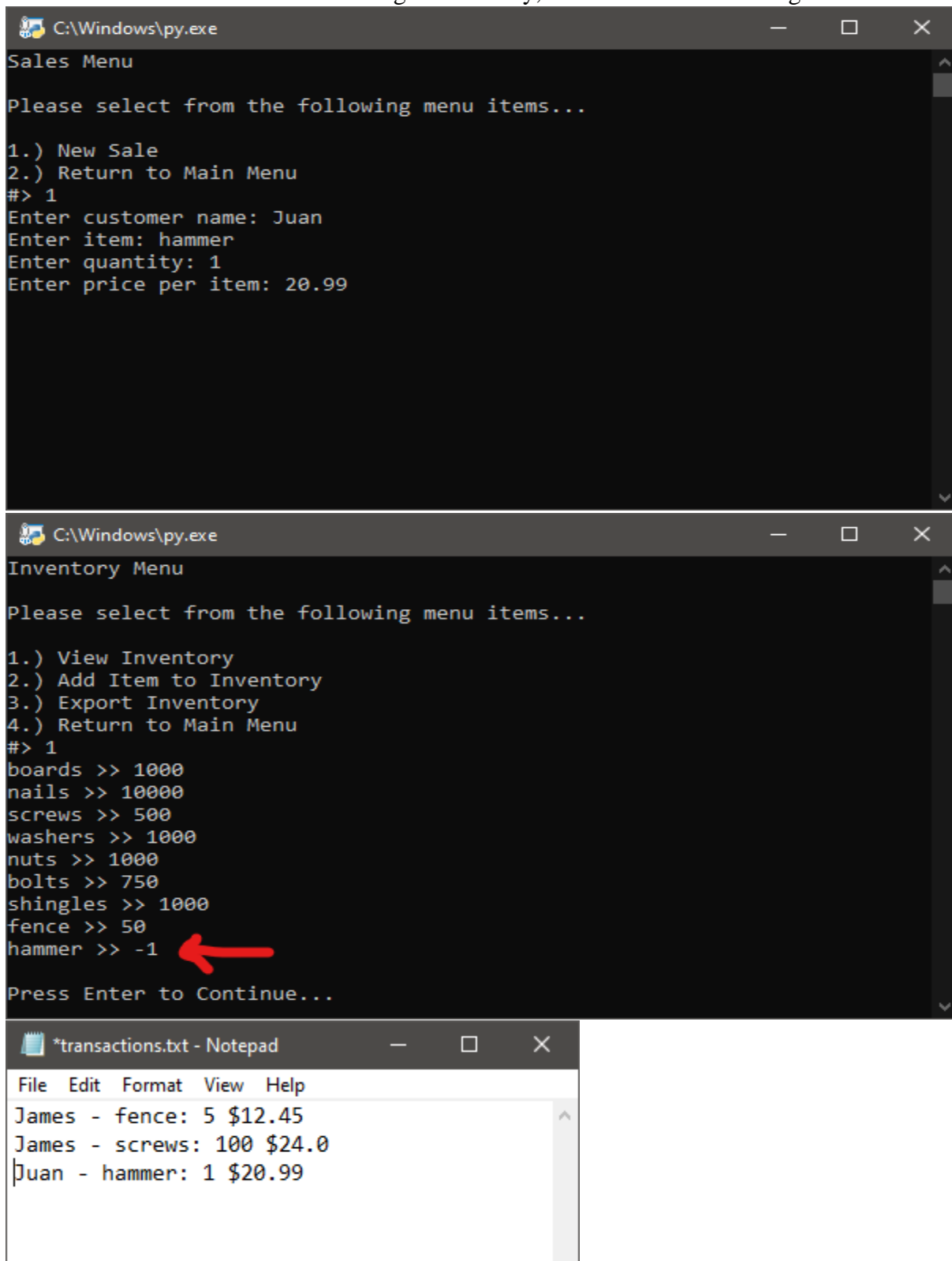
```
C:\Windows\py.exe
Inventory Menu

Please select from the following menu items...

1.) View Inventory
2.) Add Item to Inventory
3.) Export Inventory
4.) Return to Main Menu
#> 1
boards >> 1000
nails >> 10000
screws >> 500
washers >> 1000
nuts >> 1000
bolts >> 750
shingles >> 1000
fence >> 50
Press Enter to Continue...
```



R1.4 & R1.6 – Transactions and adding to inventory, transaction written to log



The image shows three overlapping Windows command prompt windows. The top window is titled 'C:\Windows\py.exe' and displays a 'Sales Menu'. It prompts the user to select from menu items: 1.) New Sale, 2.) Return to Main Menu. The user enters '1'. It then prompts for customer name (Juan), item (hammer), quantity (1), and price per item (20.99). The middle window is also titled 'C:\Windows\py.exe' and displays an 'Inventory Menu'. It prompts the user to select from menu items: 1.) View Inventory, 2.) Add Item to Inventory, 3.) Export Inventory, 4.) Return to Main Menu. The user enters '1'. It then displays a list of inventory items with their quantities: boards >> 1000, nails >> 10000, screws >> 500, washers >> 1000, nuts >> 1000, bolts >> 750, shingles >> 1000, fence >> 50, and hammer >> -1. A red arrow points to the 'hammer >> -1' line. The bottom window is titled '\*transactions.txt - Notepad' and shows the contents of the file: James - fence: 5 \$12.45, James - screws: 100 \$24.0, and Juan - hammer: 1 \$20.99.

```
C:\Windows\py.exe
Sales Menu

Please select from the following menu items...

1.) New Sale
2.) Return to Main Menu
#> 1
Enter customer name: Juan
Enter item: hammer
Enter quantity: 1
Enter price per item: 20.99

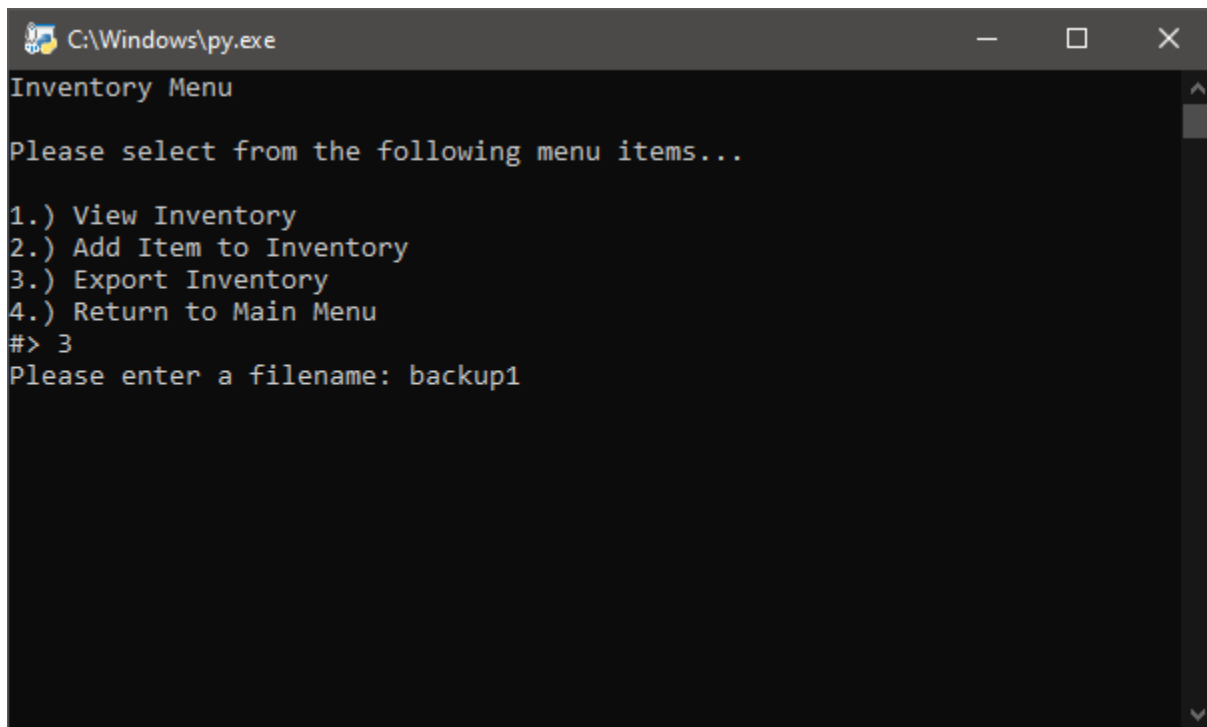
C:\Windows\py.exe
Inventory Menu

Please select from the following menu items...

1.) View Inventory
2.) Add Item to Inventory
3.) Export Inventory
4.) Return to Main Menu
#> 1
boards >> 1000
nails >> 10000
screws >> 500
washers >> 1000
nuts >> 1000
bolts >> 750
shingles >> 1000
fence >> 50
hammer >> -1
Press Enter to Continue...

*transactions.txt - Notepad
File Edit Format View Help
James - fence: 5 $12.45
James - screws: 100 $24.0
Juan - hammer: 1 $20.99
```

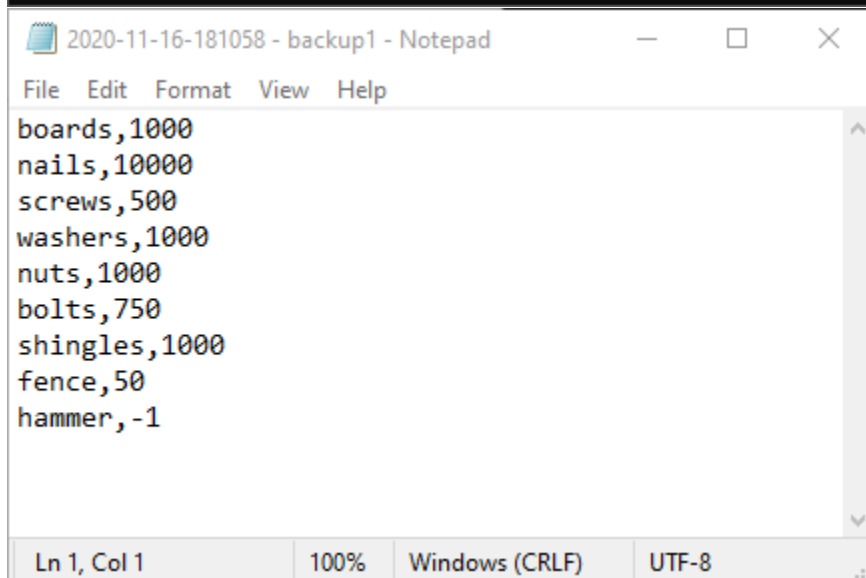
R1.5 – User requested export of inventory



```
C:\Windows\py.exe
Inventory Menu

Please select from the following menu items...

1.) View Inventory
2.) Add Item to Inventory
3.) Export Inventory
4.) Return to Main Menu
#> 3
Please enter a filename: backup1
```



```
2020-11-16-181058 - backup1 - Notepad
File Edit Format View Help
boards,1000
nails,10000
screws,500
washers,1000
nuts,1000
bolts,750
shingles,1000
fence,50
hammer,-1

Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

## R1.7 – Editable message

```
C:\Windows\py.exe
Welcome to Point of Sale/Inventory Management System
Remember to SMILE :)

Please select from the following menu items...

1.) Sales Menu
2.) Inventory Menu
3.) Settings
4.) Exit
#> 3
Settings Menu

Please select from the following menu items...

1.) Change Message of the Day
2.) Return to Main Menu
#> 1
Please enter a new message:
#> YOU CAN DO IT!
```

```
C:\Windows\py.exe
Welcome to Point of Sale/Inventory Management System
YOU CAN DO IT!
Please select from the following menu items...

1.) Sales Menu
2.) Inventory Menu
3.) Settings
4.) Exit
#>
```