Software Documentation Paper

CSC 414 – Software Design

Brett Wilson

11/17/2020

Section 1.0 – Project Description

Introduction: This documentation is for the expanded stubbing project. This program will be a very basic menu program used for some type of store. The type of store has yet to be decided.

System Overview: This program will be as simple as possible. Each option will be an item and each item will have a price. The program should add up the cost of each item selected and give you a total including tax. Considering the simplicity of the program there may be added functionality if time permits. I am considering adding a way to tweak prices of the items in the menu and adding extra options. I do not want this program to be like what you would see in an early C++ class. I am not sure exactly how to introduce more complexity at the moment, but this does not change the overall design of the program. Regardless of what I add the basic skeleton of the program will still be a shop menu with items that will allow the user to choose the items and carry a running total.

Document Overview: The purpose of this document is to detail the process of creating this simple menu program, detailing the references used to create the program, and adding any requirements, design, and testing information that may be important.

Section 2.0 – References

No references used other than class slides for this paper to make sure I got the format and content of each section correct.

Below is a link to some documentation about the switch statement in Java and a link to my project and documents on github. I am just putting this here so this section is not empty.

- W3schools.com. 2020. Java Switch. [online] Available at: https://www.w3schools.com/java/java_switch.asp.
- https://github.com/Pavion941/RepoTutorial/blob/master/Test%20Results%20Document.pdf
- https://github.com/Pavion941/RepoTutorial/commit/0169f34e43e8edf7b94d9ca7ce49818d84825e8f
- https://github.com/Pavion941/RepoTutorial/blob/master/Store.java

Section 3.0 – Requirements

Basic Function:

- This program SHALL display a store menu
- This program SHALL handle errors
- This program SHALL add subtotal
- This program SHALL exit with total

Sub Requirements:

1. This program SHALL display a store menu in the command line with items and prices.
2. This program SHALL add prices to subtotal when option within range is chosen.
3. This program SHALL display invalid input when numbers outside of range, letters, or special characters are entered.
4. This program SHALL pause after each valid option and display a message reading “Press any button to continue”
5. This program SHALL calculate a total including tax when option 10 is chosen

Product:

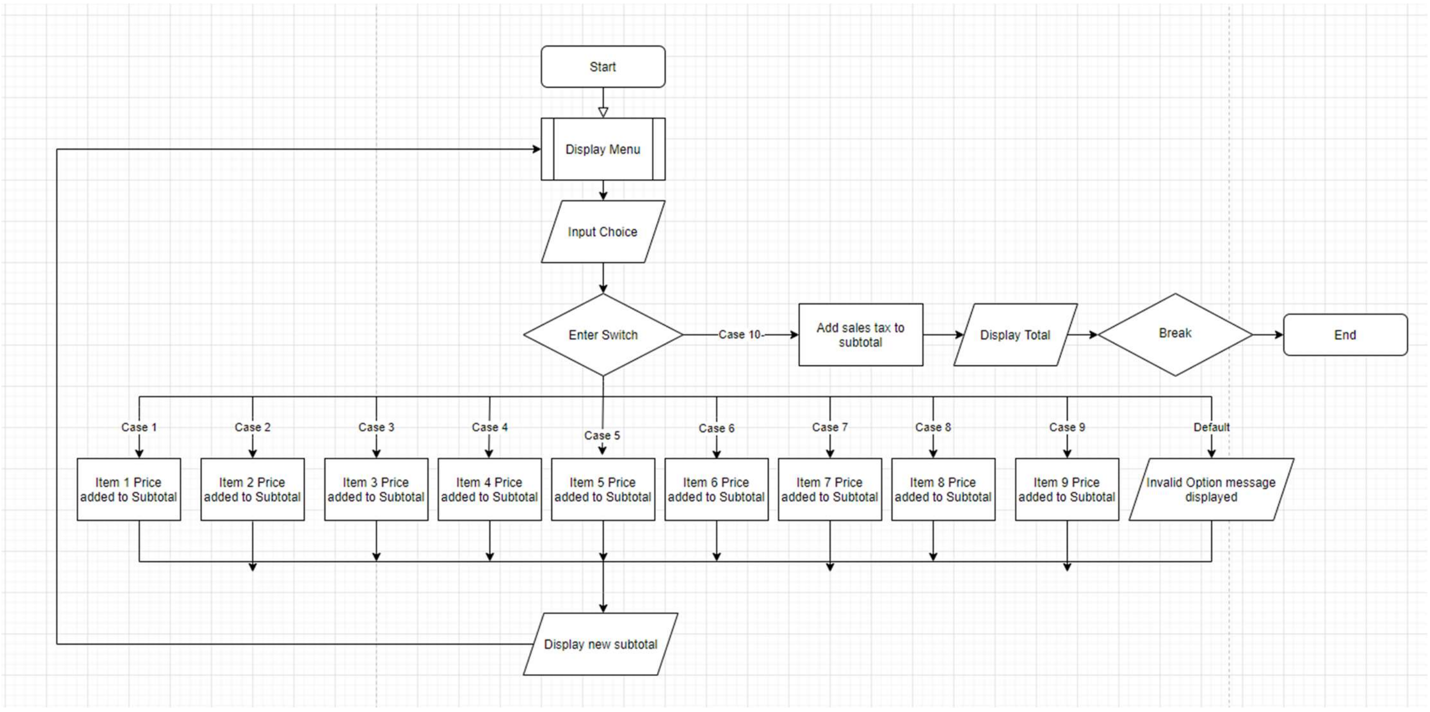
- Behavior: The menu should display in the command line when starting. The program should add to subtotal when option 1-9 are selected. When 10 is selected the program should total, add tax, and terminate. Between each valid input the program should pause and allow the user to easily read the subtotal and press any button to continue. The program should handle errors for any invalid input including numbers out of range, letters, or special characters. After an error, the program should redisplay the menu.

- Inputs: The inputs should include the choice the user makes on the menu and also the button the user presses to continue after the pause.

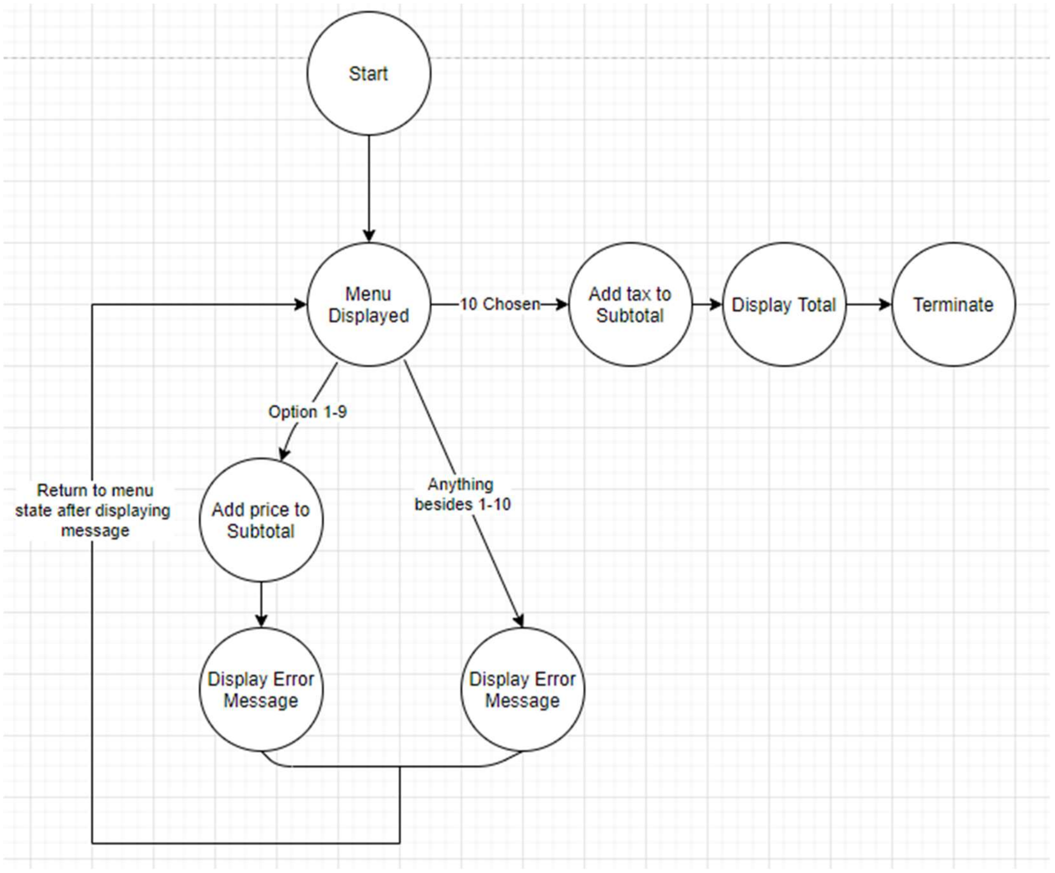
- Outputs: The outputs should include the subtotal and total.

- Processes:

- Flow Chart:


Start
Display Menu
Input Choice
Enter Switch
Case 10
Add sales tax to subtotal
Display Total
Break
End
Case 1
Case 2
Case 3
Case 4
Case 5
Case 6
Case 7
Case 8
Case 9
Default
Item 1 Price added to Subtotal
Item 2 Price added to Subtotal
Item 3 Price added to Subtotal
Item 4 Price added to Subtotal
Item 5 Price added to Subtotal
Item 6 Price added to Subtotal
Item 7 Price added to Subtotal
Item 8 Price added to Subtotal
Item 9 Price added to Subtotal
Invalid Option message displayed
Display new subtotal

- State Diagram:

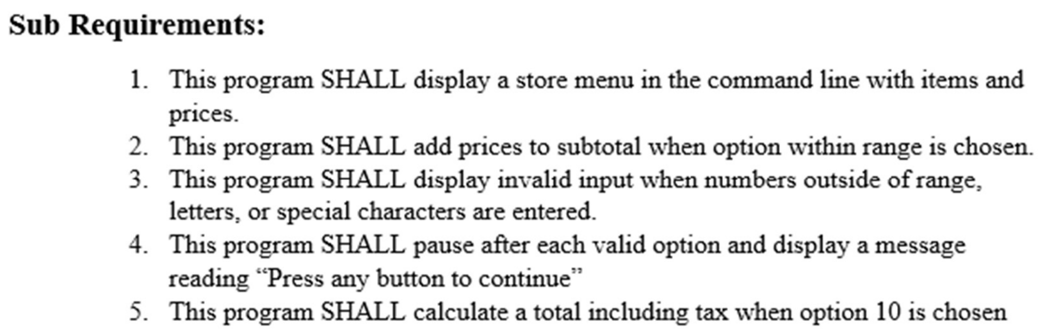


Section 4.0 – Design

This document details the ways in which I will satisfy the requirements listed in Section 3.0

I will explain the ways to tackle the sub requirements in the same order they are listed in Section 3.0.

For clarity and easy reference, I will list the sub requirements below.



Sub Requirements:

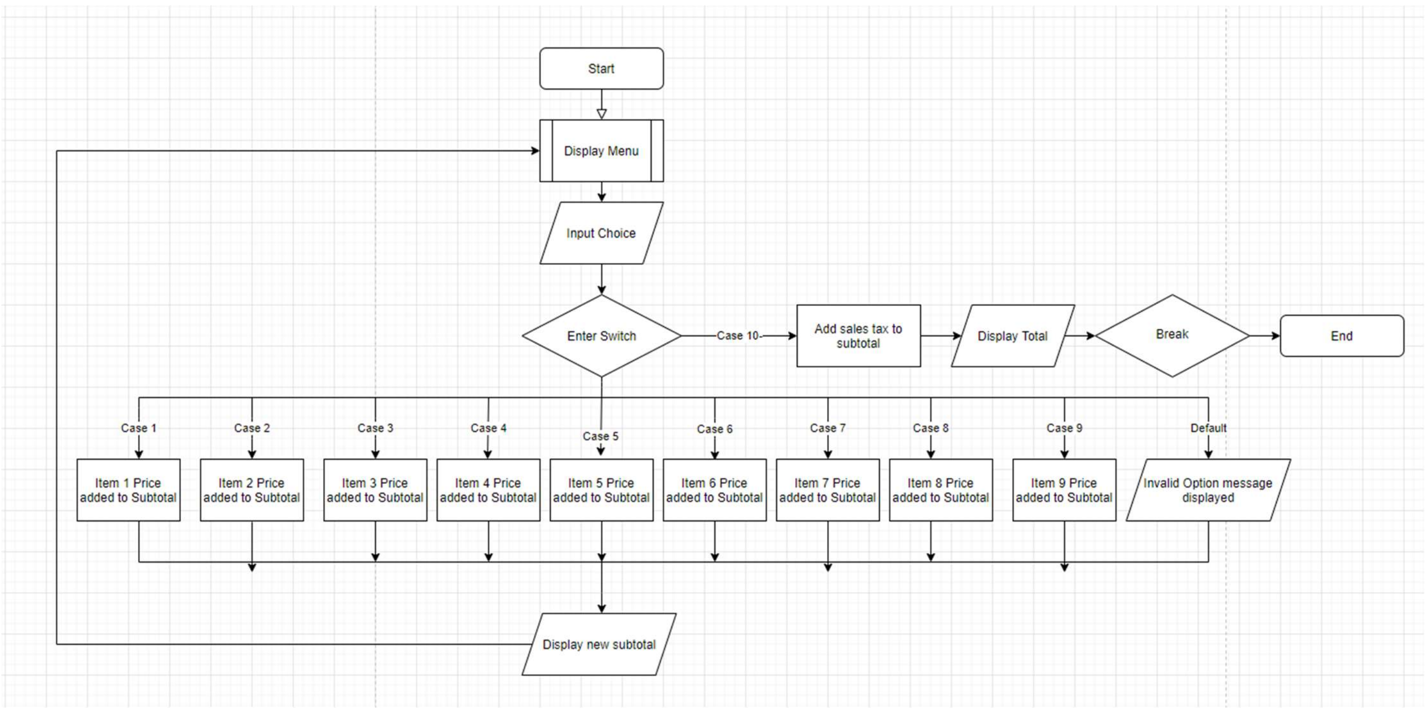
1. This program SHALL display a store menu in the command line with items and prices.
2. This program SHALL add prices to subtotal when option within range is chosen.
3. This program SHALL display invalid input when numbers outside of range, letters, or special characters are entered.
4. This program SHALL pause after each valid option and display a message reading "Press any button to continue"
5. This program SHALL calculate a total including tax when option 10 is chosen

Design choices:

1. Using a do while loop and "printf" statements I will print the menu options along with instructions for the user. I will then use "input.nextInt()" to store the users choice in a choice variable. I will handle errors with invalid datatype input using a try and catch statement.
2. Using a switch statement, I will take the user input, stored in choice, and add the appropriate amount to the subtotal for the item chosen. This should only work when the number stored in choice is within range of the menu options and valid.
3. Using the switch statements default, I will handle user input that is not in range of the menu options. The previously mentioned try and catch statement will catch letters or special characters entered by the user.
4. Using a new variable called "pause", I will insert a new "printf" and "input.nextInt()" into each case of the switch statement so when they are selected the user will be forced to "Press any button to continue." I will also handle invalid data type input errors by inserting a try and catch statement. This will also give the user the option to press any button on their keyboard to continue with the program.
5. Using a variable called total I will add sales tax to the subtotal and display it when option 10 is chosen. I will then break out of the do while loop because the condition will be satisfied (choice = 10).

For clarity I will add the flow chart from section 3.0 to the Design section as well. I initially thought that you told us to add it to the design part but when I read over the notes it was mentioned in the requirements part.

- Flow Chart


Start
Display Menu
Input Choice
Enter Switch
Case 10
Add sales tax to subtotal
Display Total
Break
End
Case 1
Case 2
Case 3
Case 4
Case 5
Case 6
Case 7
Case 8
Case 9
Default
Item 1 Price added to Subtotal
Item 2 Price added to Subtotal
Item 3 Price added to Subtotal
Item 4 Price added to Subtotal
Item 5 Price added to Subtotal
Item 6 Price added to Subtotal
Item 7 Price added to Subtotal
Item 8 Price added to Subtotal
Item 9 Price added to Subtotal
Invalid Option message displayed
Display new subtotal

Section 5.0 – Test Plan

Required Test

- Run the program and visually confirm working store menu
- Enter various characters into the menu selection and check for errors
- Enter various characters into pause function and check for errors
- Choose multiple options and check for errors including subtotal accuracy
- Choose exit and check for errors including total accuracy

Test Procedure

- Compile, run, and visually inspect the program to make sure a menu is displayed in command line.
    - Expected outcome: Menu with 9 items and 1 exit option is displayed with no errors.

- Enter a number, letter, and special character into the menu selection and check the results of each type of character for errors.
    - Expected outcome: Numbers inside the range of the menu should select the appropriate option and display the correct subtotal. Numbers outside the range of the menu should kick you back to selection. Letters and special characters should throw an "Invalid input" error message and kick you back to selection

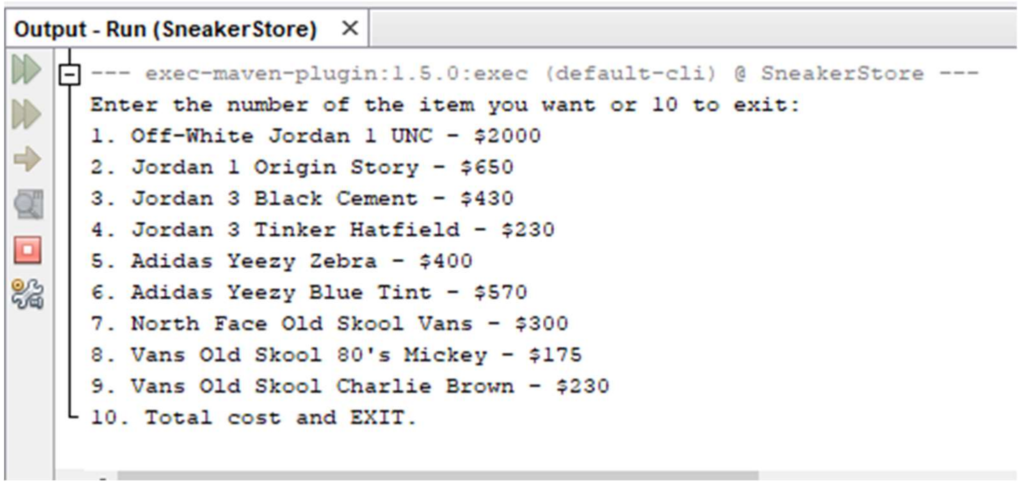
- Enter a number, letter, and special character into the pause function and check the results of each type of character for errors.
    - Expected outcome: Numbers, letters, and special characters should pass without error and trigger the menu to reopen in command line.

- Choose multiple items within the range of the menu and check if they all work without error. Also check if the options entered are adding up to the correct subtotal amount.
    - Expected outcome: The options selected will all work without error and with the correct pricing. Each time a new item is selected, that item's cost will be added to the subtotal.

- Choose the exit option and check for errors. Also check if the total, including tax, calculated at termination of the program is accurate to the expected total.
    - Expected outcome: The exit option should calculate the total, including tax, and output the correct total amount.

Appendix – Test Results

Test 1: Visual inspection of menu

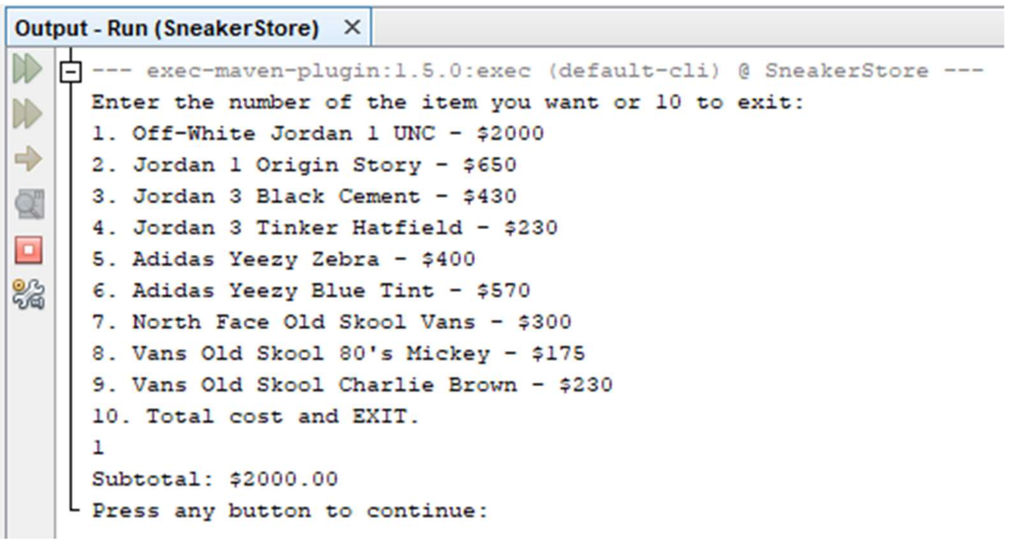


Output - Run (SneakerStore)

```
--- exec-maven-plugin:1.5.0:exec (default-cli) @ SneakerStore ---
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
```

Results/Conclusion: On inspection the menu is displayed properly with the correct prices.

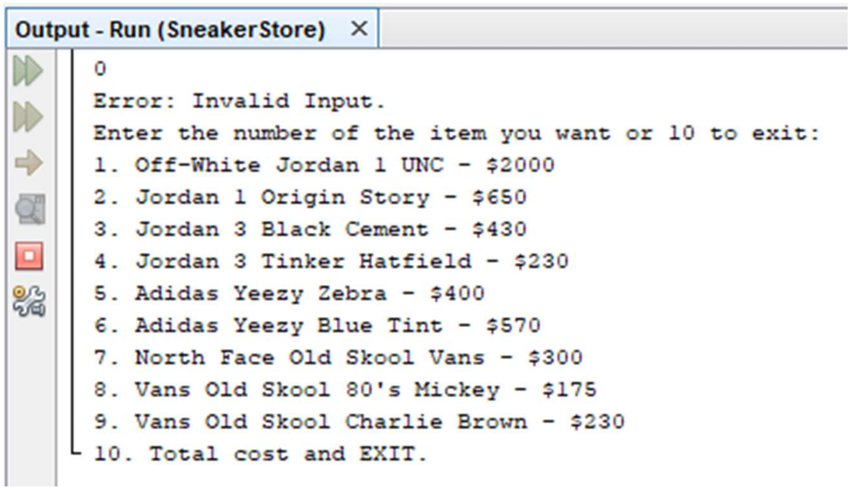
Test 2: Enter various characters into selection and check for errors



Output - Run (SneakerStore)

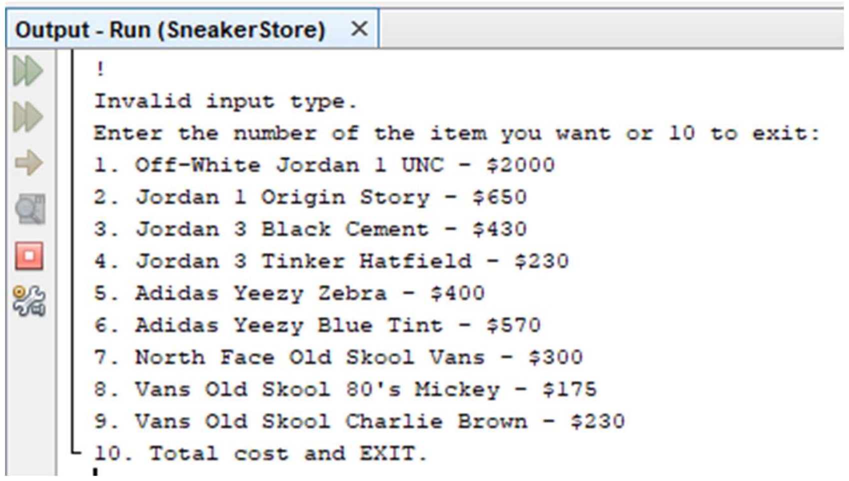
```
--- exec-maven-plugin:1.5.0:exec (default-cli) @ SneakerStore ---
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
1
Subtotal: $2000.00
Press any button to continue:
```

Results/Conclusion: Entering a number within range of the menu gives the correct dollar amount subtotal without errors. All tested and working properly.



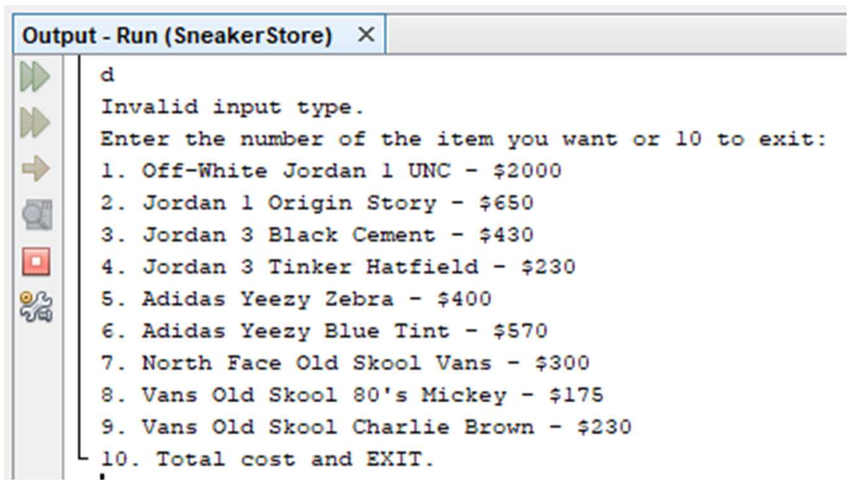
```
0
Error: Invalid Input.
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
```

Results/Conclusion: Entering a number outside the range of the menu gives the correct "Invalid Input" display and reopens the menu.



```
!
Invalid input type.
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
```

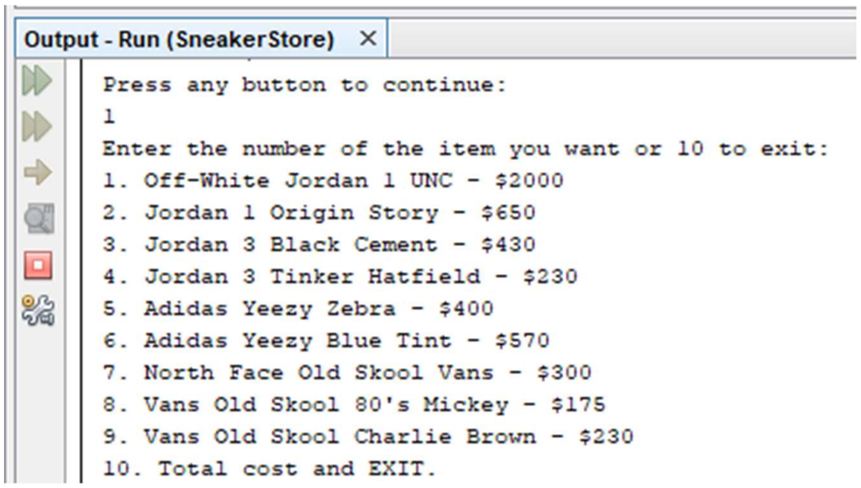
Results/Conclusion: Entering a special character gives the correct "Invalid Input" display and reopens the menu.



```
d
Invalid input type.
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
```

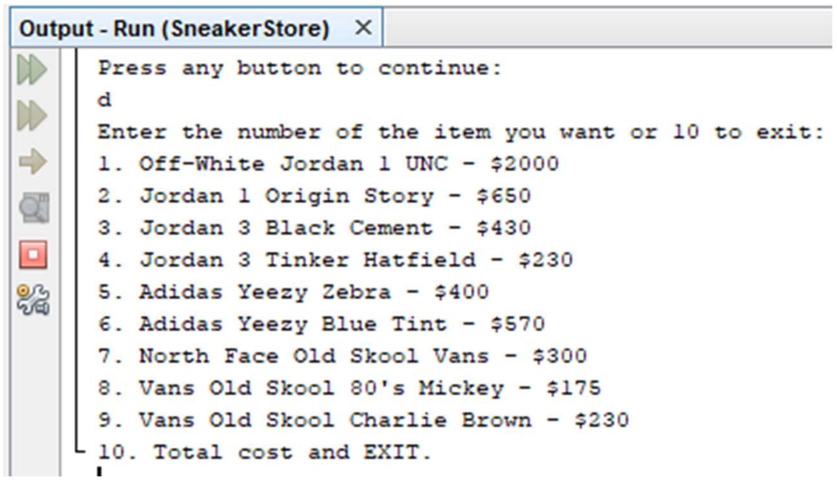
Results/Conclusion: Entering a letter gives the correct "Invalid Input" display and reopens the menu.

Test 3: Enter various characters into pause function and check for errors



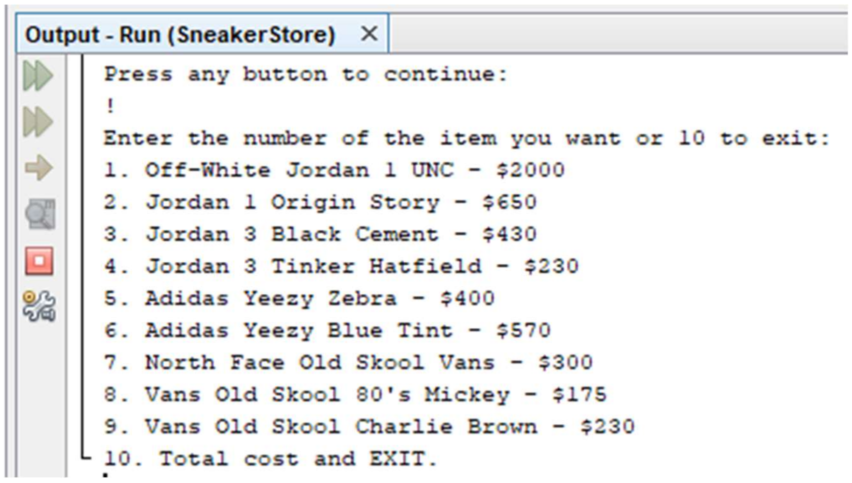
```
Press any button to continue:
1
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
```

Results/Conclusions: Entering a number when prompted will reopen the menu



```
Press any button to continue:
d
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
```

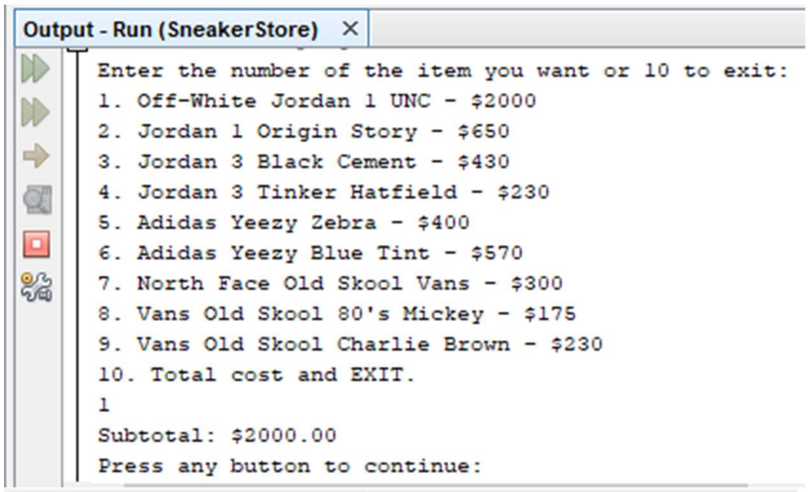
Results/Conclusions: Entering a letter when prompted will reopen the menu



```
Press any button to continue:
!
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
```

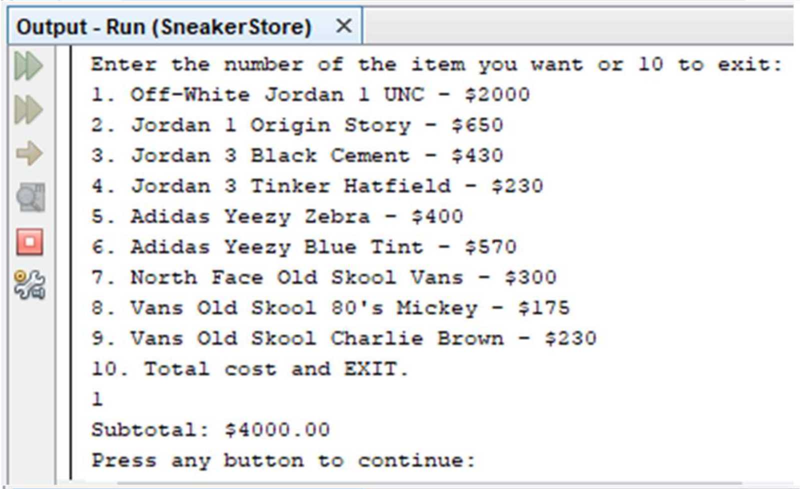
Results/Conclusions: Entering a special character when prompted will reopen the menu

Test 4: Choose multiple options and check for errors and accuracy



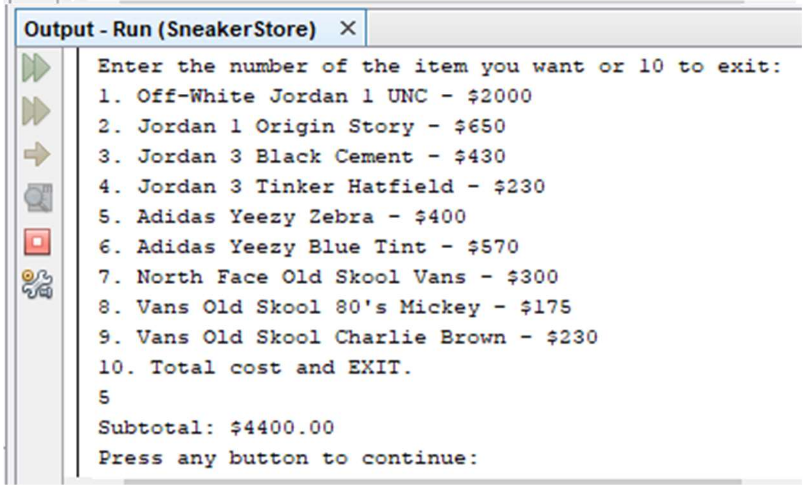
Output - Run (SneakerStore)

```
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
1
Subtotal: $2000.00
Press any button to continue:
```



Output - Run (SneakerStore)

```
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
1
Subtotal: $4000.00
Press any button to continue:
```

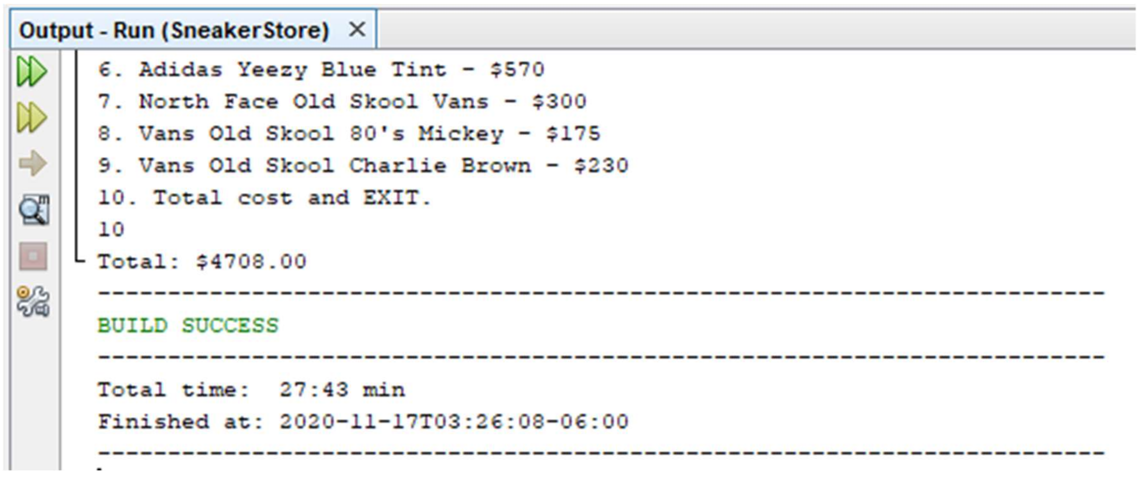


Output - Run (SneakerStore)

```
Enter the number of the item you want or 10 to exit:
1. Off-White Jordan 1 UNC - $2000
2. Jordan 1 Origin Story - $650
3. Jordan 3 Black Cement - $430
4. Jordan 3 Tinker Hatfield - $230
5. Adidas Yeezy Zebra - $400
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
5
Subtotal: $4400.00
Press any button to continue:
```

Results/Conclusion: Multiple options add accurately to subtotal

Test 5: Choose exit and check for errors and accuracy



Output - Run (SneakerStore)

```
6. Adidas Yeezy Blue Tint - $570
7. North Face Old Skool Vans - $300
8. Vans Old Skool 80's Mickey - $175
9. Vans Old Skool Charlie Brown - $230
10. Total cost and EXIT.
10
Total: $4708.00
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  27:43 min
Finished at: 2020-11-17T03:26:08-06:00
------------------------------------------------------------------------
```

Results/Conclusions: Option 10 correctly exited the program while adding sales tax to the final total.