**Patrick A. Pangilinan BSCS – C2A**

**SA 3 Program Documentation**

1. Discuss the structure of your code. Enumerate the different control structures that you used and in what aspect of the problem did you use them.

* I structured my code around the Object-Oriented Programming structure, by detailing the methods and variables that I used around the class that I instantiated. The first parts that the program goes through is the menu which uses a switch case control structure to choose between the customer, admin menu or to exit the program.
* Once the user goes to the admin menu, another switch case is used to choose between showing inventory, sales, purchases, out of stock items or go back to the main menu. The inventory, sales and purchases make use of if statements for styling using indents if the loop reaches a certain condition to make the table coherent and readable. This goes for the rest of the functions that use the same principle.
* Showing the out-of-stock items table makes use of switch case statement to choose whether to restock an item or go back to the menu. Once the user chooses to restock an item, the program then uses if else statements in a loop to check whether there are out of stock items in the inventory. If there are no out of stock items, the program will not allow the user to restock any item.
* Going back to the main menu, if the user chooses the customer menu, the user will be given a choice to check items, buy items or go back using a switch case statement. If the user chooses to buy items, the user will choose a product ID which will then check using if else statements if the product chosen is out of stock or not as well as checking the input whether the product ID is below 1 or above 15. The quantity input is also checked with an if else statement to see whether or not the input is below 0 or above the available quantity of the item.
* The bottom line is that the majority of the control structures I used in my program are the switch case, if else statements and looping control structures.

1. Provide an inventory of your code. It will include how many libraries you used, variables, functions, classes, objects.

|  |  |  |
| --- | --- | --- |
| Libraries | Classes | Objects |
| <iostream> | Inventory | WSInventory |
| <fstream> |  |  |
| <string> |  |  |
| <ctime> |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variables | | | | |
| int | **long double** | **char** | **string** | **bool** |
| pID | price | choice | invMaster[16][8] | stop |
| id | restockPrice |  | outOfStockMaster[16][8] |  |
| qty |  |  | line |  |
| invQty |  |  |  |  |
| restockQty |  |  |  |  |
| newQty |  |  |  |  |
| sQty |  |  |  |  |
| rop |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Functions | | | |
| GUI Functions | **Array Initializer Functions** | **Data Printer** | **Data Updater** |
| void menu() | void loadInventory() | void showInventory() | string getTime() |
| void adminMenu() | void loadOutofStock() | void showSales() | void updatePurchases() |
| void customerMenu() | void checkOutofStock() | void showPurchases() | void updateSales() |
| void restockItems() |  | void showOutofStock() | void updateInventory() |
| void buyItems() |  |  |  |
| bool retry() |  |  |  |

1. If you declared functions, enumerate the functions and provide a short description for each function.

void menu()

-Shows the main menu and the options to pick between the admin, customer menu or exit the program.

void adminMenu()

-Give the user the options to show inventory, sales, purchases, out of stock items or to go back to the main menu.

void customerMenu()

-Gives the user the options to check the items in the inventory, buy items or go back to the main menu.

void restockItems()

-Shows the out of stock list, and the option to reorder out of stock items if there are out of stock items in the list or go back to admin menu.

void buyItems()

-Shows the inventory and lets the customer choose the product id and quantity to buy an item.

bool retry()

-Facilitates whether the user retries a certain procedure or not by inputting ‘Y’ or ‘N’.

void loadInventory()

- Intializes the inventory array by taking the data from the inventory master file.

void loadOutofStock()

- Intializes the out of stock master array by taking the data from the out of stock file.

void checkOutofStock()

- Intializes the inventory array by taking the data from the inventory master file.

void showInventory()

-Prints the inventory master array.

void showSales()

-Prints the sales text file.

void showPurchases()

-Prints the purchases text file.

void showOutofStock()

-Prints the out of stock master array.

string getTime()

-Gets the current date for reorders or sales.

void updatePurchases()

-Updates the purchases file by appending new reorders or purchases.

void updateSales()

-Updates the sales file by appending new sales.

void updateInventory()

-Updates the inventory after a reorder or sale.

1. If you used OOP, identify the principles of OOP that you have used (E.g. classes, objects, inheritance, and others) and discuss the use of those principles in your code.

* I used OOP by creating a class to handle all the functions that were carried out in the wholesale inventory management system. The class being named “Inventory”, and the object named “WSInventory”. The methods I used above were all contained by this class.
* I used encapsulation by restricting direct access to some of the components of the methods that I used such as variables that are only related to that one method. Abstraction was also used in the program by hiding the parts that the user doesn’t necessarily need to see, hence why the use of public and private methods and why the menu is the only public method that I used in the program.
* Lastly, I used the concept of recursion on most of the functions to facilitate the repeat of a certain method in case the user input is invalid or to repeat a certain action in an incrementing fashion.