**Name: Md Rakeen Murtaza**

CS 1337.012 Timothy McMahan

Project 4 Pseudocode

In the main function:

Declare variables:

* strings: userOption, relateUserOption, userChoice, all the variables to hold user information
* bool: adminLoginStatus
* int: inventoryItemPosition = -1, cartItemPosition = -1

Open a text file named “ProductData.csv” for both input and output. Test for file open errors.

if ( dataFile.fail() )

Print “File could not be opened successfully”

else

Declare a dynamically allocated inventory object:

* Inventory\* todaysInventory = new Inventory(dataFile)

Put the entire main program in a do-while loop

do

* Print options and prompt for user input

“Choose an Option:

1 – Admin or A – Admin

2 – Customer or C – Customer

3 – Exit or E – Exit”

* Get user input and store in userOption
* Use toupper function to change userOption and store it in relateUserOption
* while (relateUserOption != “ADMIN” && relateUserOption != “A” && relateUserOption != “C” && relateUserOption != “CUSTOMER” && relateUserOption != “EXIT” && relateUserOption != “E”)
  + Print error message
  + Prompt for new input. Store in userOption.
  + Use toupper function to change userOption and store it in relateUserOption
* while (relateUserOption == “ADMIN” || relateUserOption == “A”)
  + Dynamically declare an Admin object: Admin\* rakeen = new Admin(rakeen01, 82aaBB!)
  + adminLoginStatus = rakeen.getLoginInfo()
  + if (adminLoginStatus)

“Choose an Option:

* + - * Add Product
      * Delete Product
      * Update Qty
      * Update Cost

while (invalid choice)

Print error message and validate input

if (add product)

Get itemNumber

Get itemName

rakeen.adminChoice.addItem(itemNumber, itemName)

else if (delete product)

Get itemName

rakeen.adminChoice.deleteItem(itemName)

else if (update qty)

Get new qty

rakeen.adminChoice.setQty(newQty)

else if (update cost)

Get new cost

rakeen.adminChoice.setCost(newCost)

* + else

Print “Invalid login credentials”

Print options and prompt for user input

“Choose an Option:

1 – Admin or A – Admin

2 – Customer or C – Customer

3 – Exit or E – Exit”

Get user input and store in userOption

Use toupper function to change userOption and store it in relateUserOption

while (relateUserOption != “ADMIN” && relateUserOption != “A” && relateUserOption != “C” && relateUserOption != “CUSTOMER” && relateUserOption != “EXIT” && relateUserOption != “E”)

* + - * Print error message
      * Prompt for new input. Store in userOption.
      * Use toupper function to change userOption and store it in relateUserOption
  + delete rakeen
  + rakeen = nullptr

end while for admin

* while (relateUserOption == “CUSTOMER” || relateUserOption == “C”)

Dynamically declare a customer object:

Customer\* customer1 = new Customer();

Call customer1.createInvoiceFile() function

do

Print “Do you want to:

* + - Add item to the shopping cart?
    - Delete item from the shopping cart?
    - Update Qty of item in the shopping cart?
    - Display the shopping cart?
    - Finish shopping?

Get user’s choice

while (invalid option chosen)

print error message

Get user input again

if (user has not finished shopping)

Get item name and store it in userChoice

if (add item)

customer1.addItem(userChoice)

else if (deleteItem)

customer1.deleteItem(userChoice)

else if (updateQty)

customer1.updateQty(userChoice)

else if (show cart)

customer1.showCart()

while (user has not finished shopping)

customer1.updateInvoiceFile(lastName.txt)

* + delete customer1
  + customer1 = nullptr

Print options and prompt for user input

“Choose an Option:

1 – Admin or A – Admin

2 – Customer or C – Customer

3 – Exit or E – Exit”

Get user input and store in userOption

Use toupper function to change userOption and store it in relateUserOption

while (relateUserOption != “ADMIN” && relateUserOption != “A” && relateUserOption != “C” && relateUserOption != “CUSTOMER” && relateUserOption != “EXIT” && relateUserOption != “E”)

* + - * Print error message
      * Prompt for new input. Store in userOption.
      * Use toupper function to change userOption and store it in relateUserOption

end while for customer

todaysInventory.editProductData(datafile)

while (relateUserOption != “EXIT” && relateUserOption != “E”)

delete todaysInventory

todaysInventory = nullptr

**Overloaded += operator:**

void operator+=(node TempStruct)

appendNode(headRef, TempStruct)

// headRef is a member of the LinkedList class which will be on the left side of the += operator

// TempStruct is a struct variable containing all information about the new product, which will be on the right side of the += operator