Development Diary

By: Lavani Somesan

# Identifiable Classes

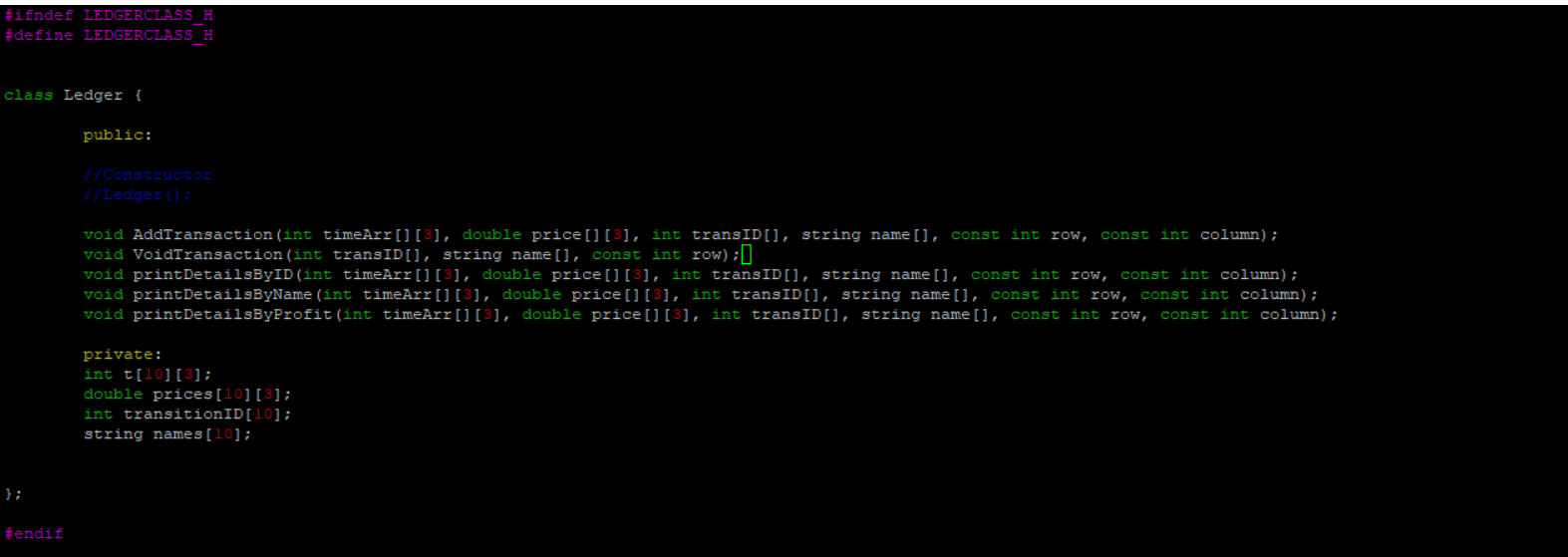
* Ledger Class (Holds Adding transactions, Voiding transactions, and Printing transactions)
* **Functions of Ledger Class**
* Add Transaction
* Void a Transaction
* Prints Summary of Transaction based on different sorting methods

# Steps to Implement Code

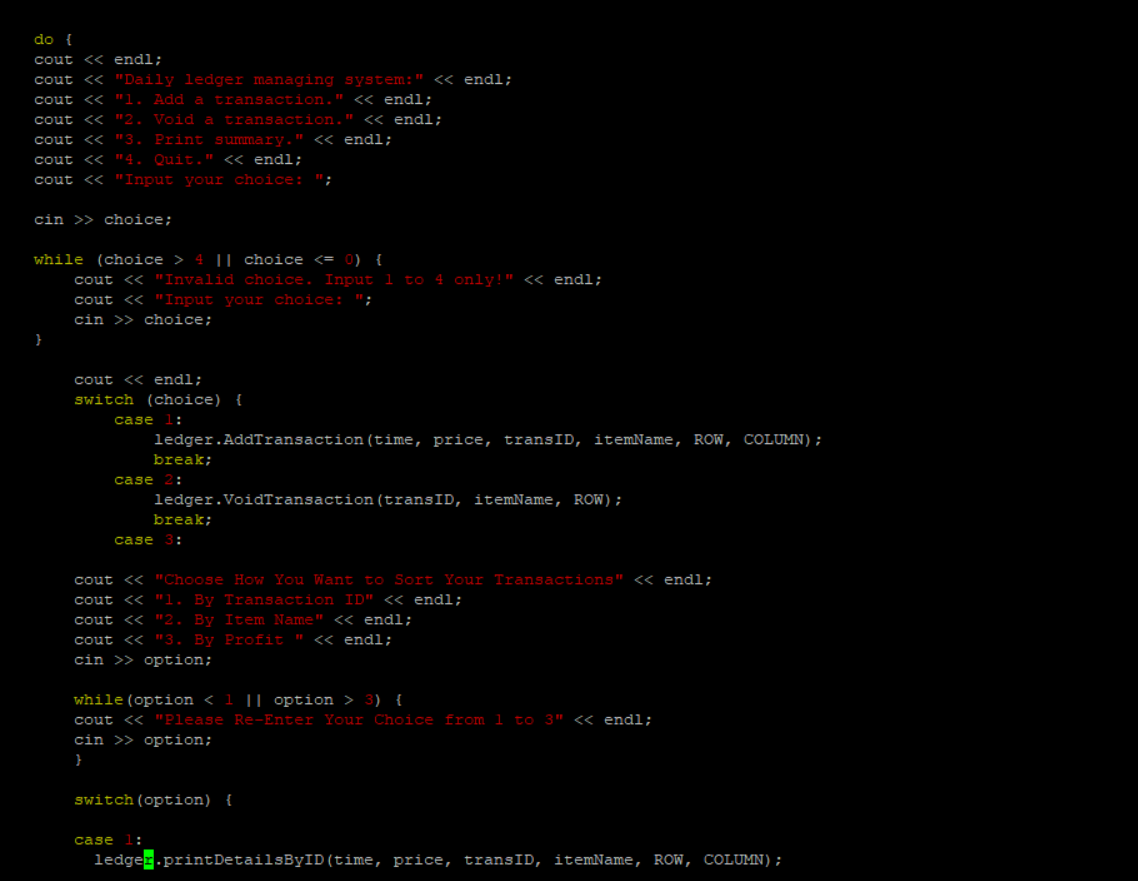
(There was a starter code pack but I chose to do my code project from scratch)

1. Create the ledger.h file which will hold the class declaration of the ledger class. It will contain several functions as mentioned above.
2. Next Create the main program but exclude the class instantiation and functions calls from main program.
3. Create the ledger.cpp file which will hold the ledger class declaration. Declare the first function (AddTransaction) but don’t write anything in it. Link class to main program and compile to see if you get any errors then proceed to write function if no errors. Then Run to see if it works.
4. Next define and declare VoidTransaction function and link it to main program. Compile to see if you get an error. If not, then proceed to write the function and then compile again to see if it runs correctly.
5. Next, define and declare printDetails function and link it to main program. Compile to see if you get an error. If not, then proceed to write the function and then compile again to see if it runs correctly. Run Program to see if it works correctly

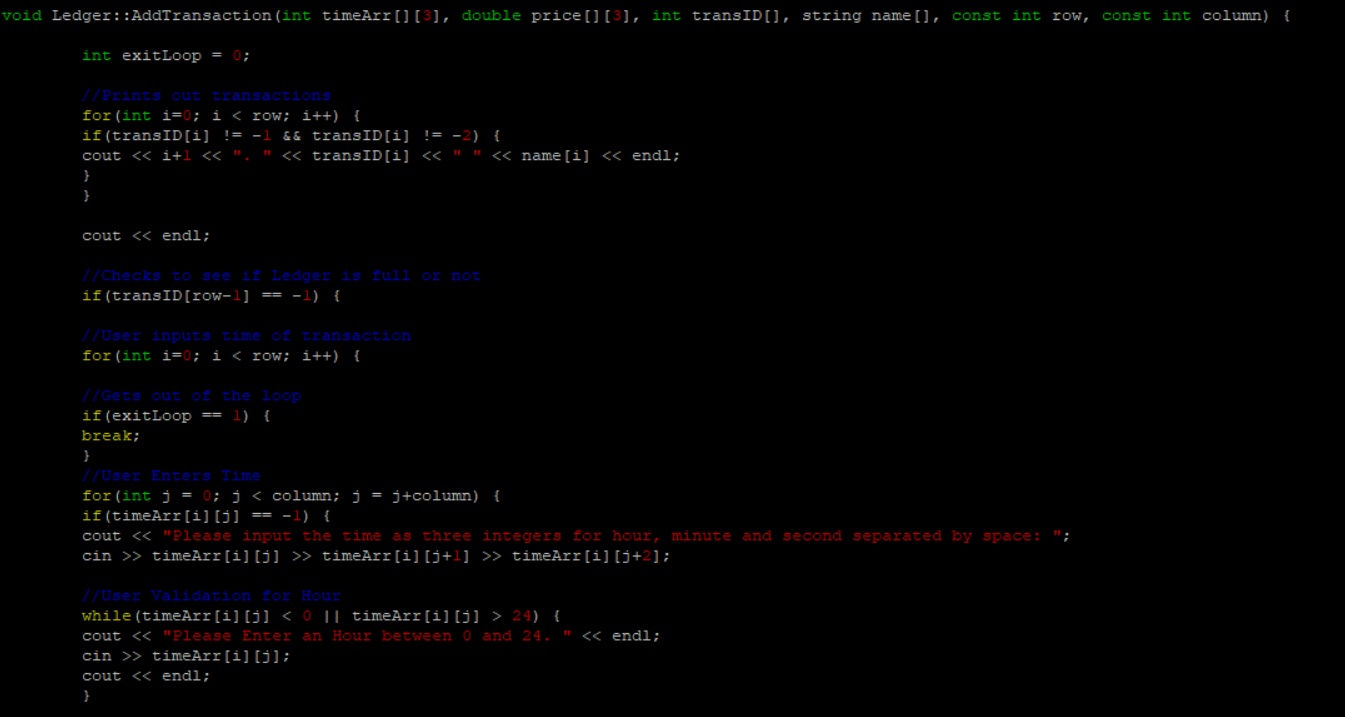
# Step 1

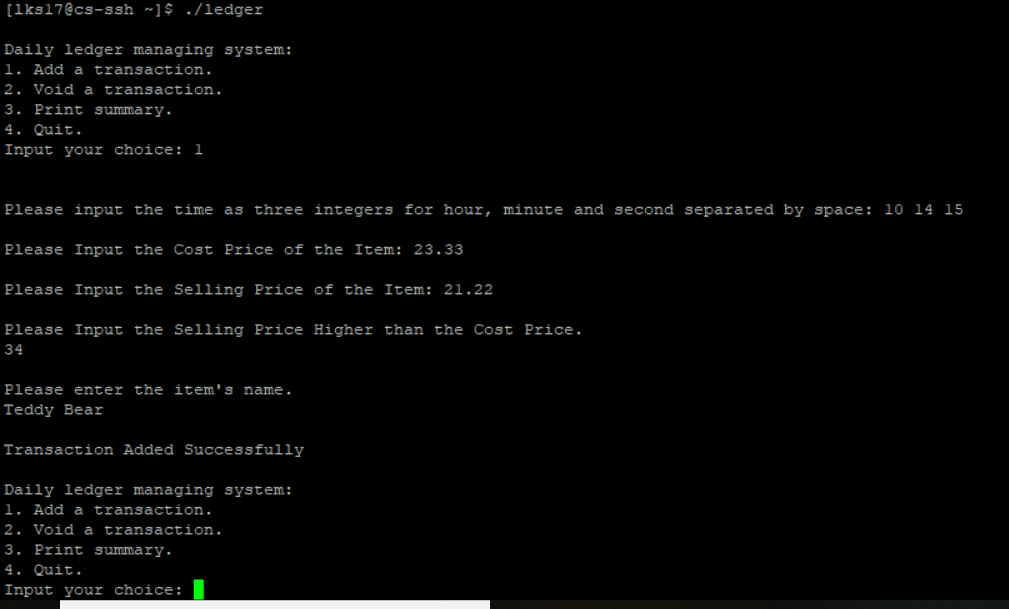


# Step 2:

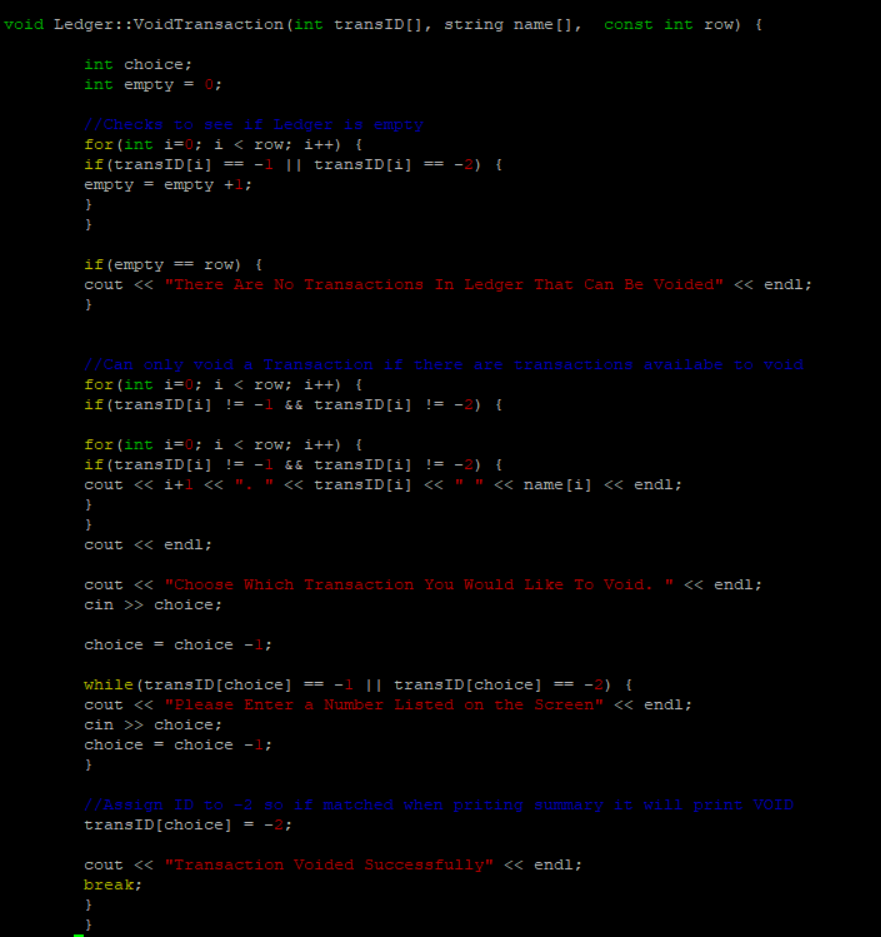


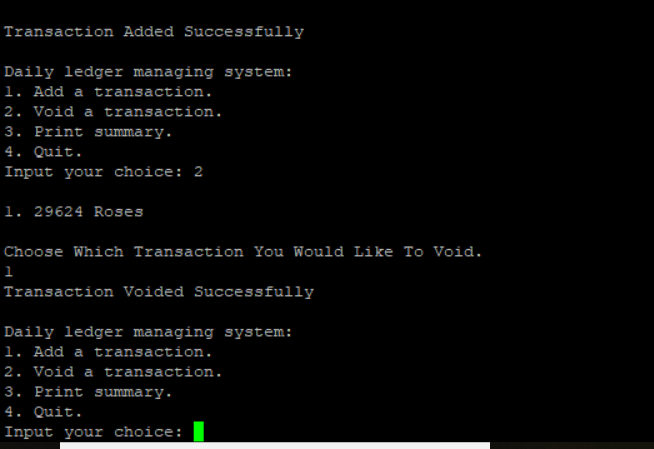
# Step 3:



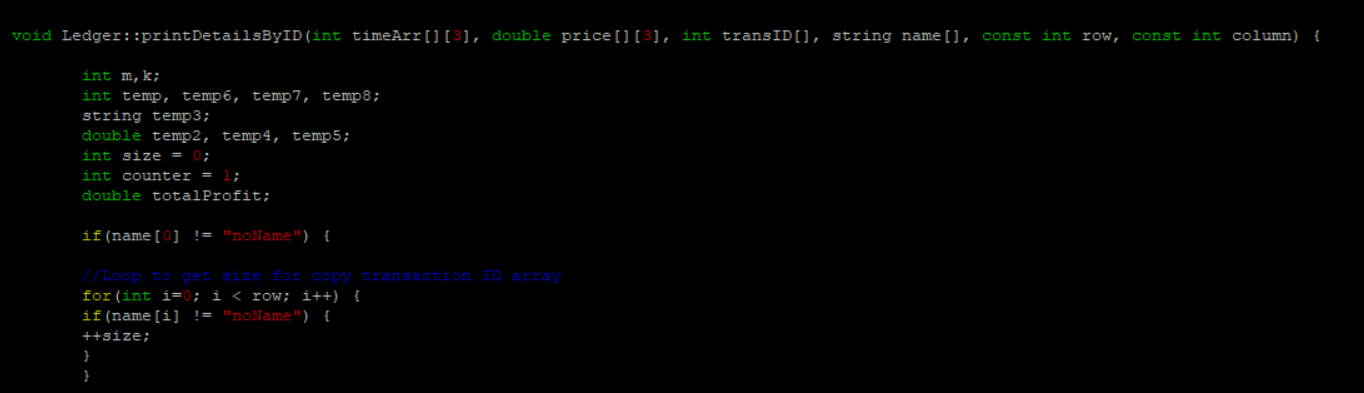


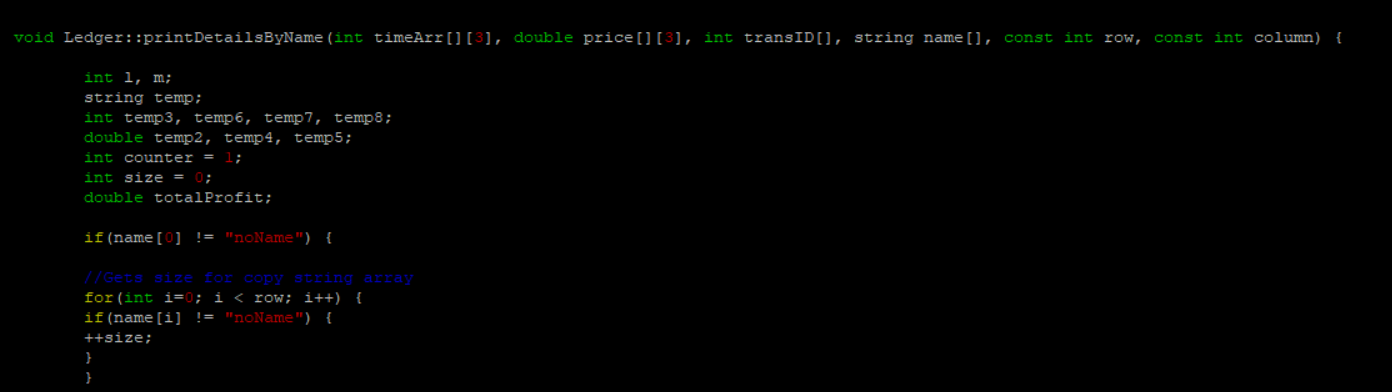
# Step 4:

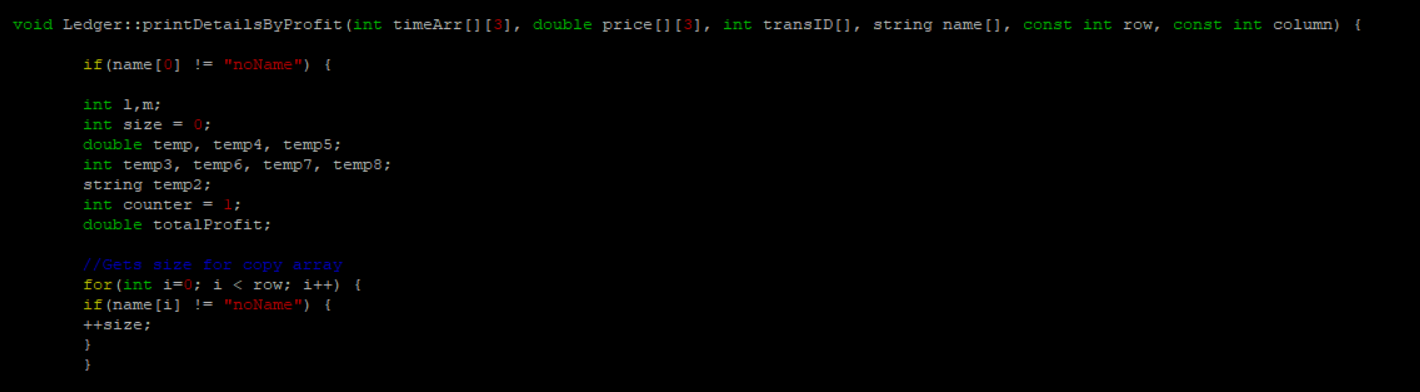


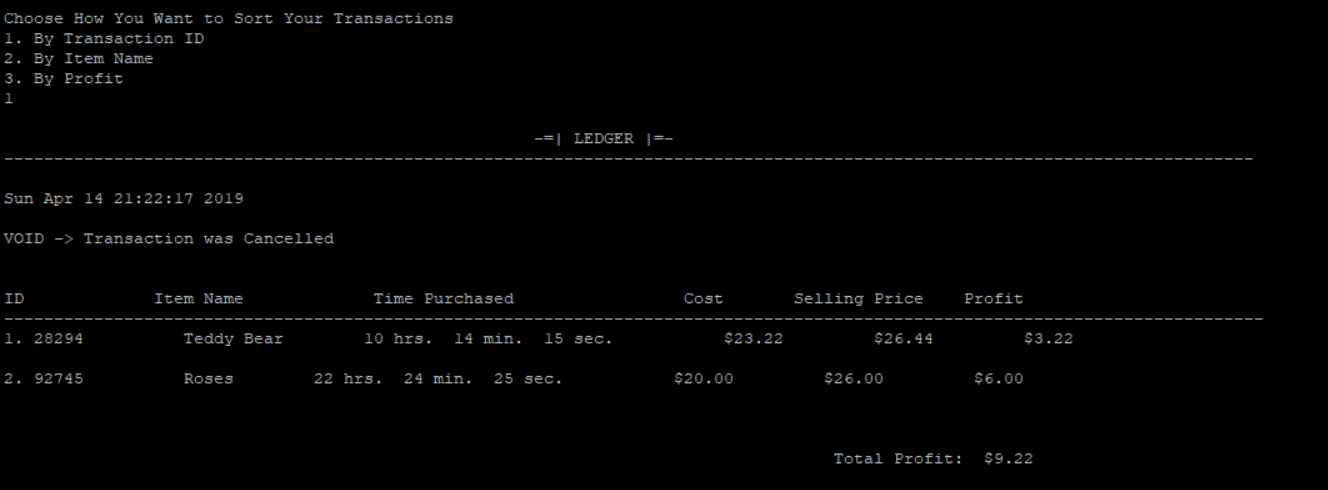


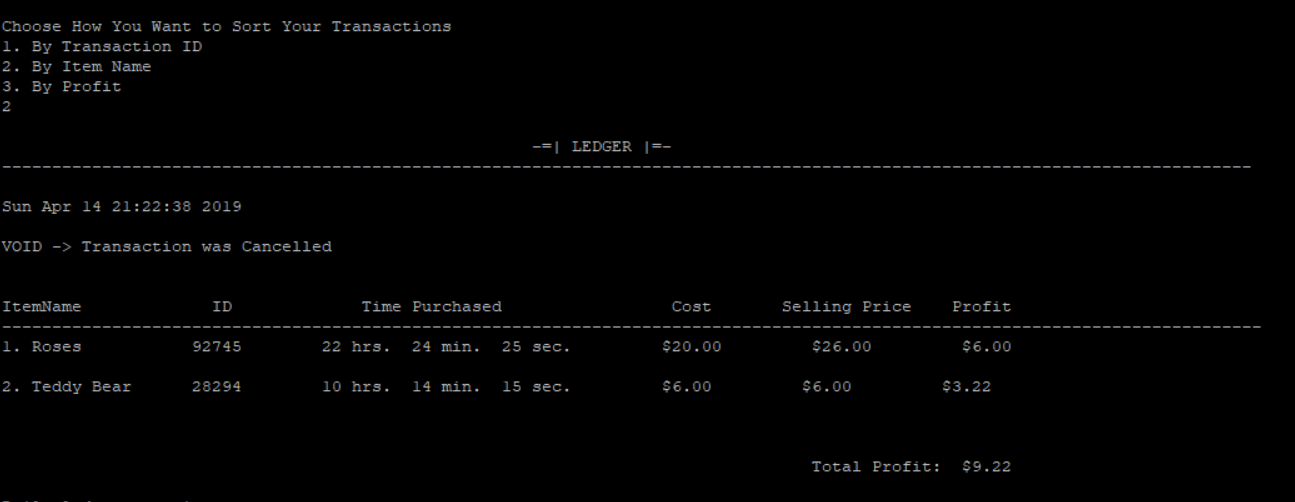
# Step 5:

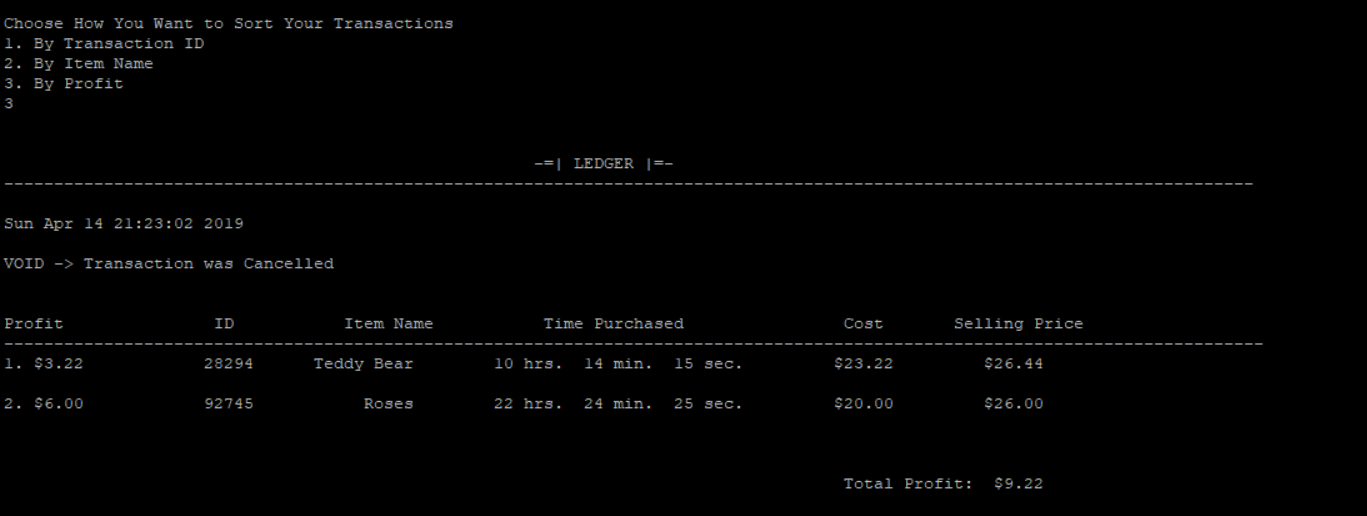












# Reflection:

I originally was only going to have one printDetails Function but for some reason something went wrong when putting all the different sorting arrays together in different cases. So, I had to take the menu and put it into main and create different printDetails by how you wanted the ledger sorted. I only had to use one class like I intended originally since there weren’t any extra functions that were needed from other classes. I also did not use the starter code pack, I did my code from scratch. The language used in the project was unclear at times so I assumed that when it meant deleting/removing a transaction it meant to VOID it. Which it said in the project description when user wants to cancel a transaction to append VOID to it therefore canceling it but not removing it from the ledger.