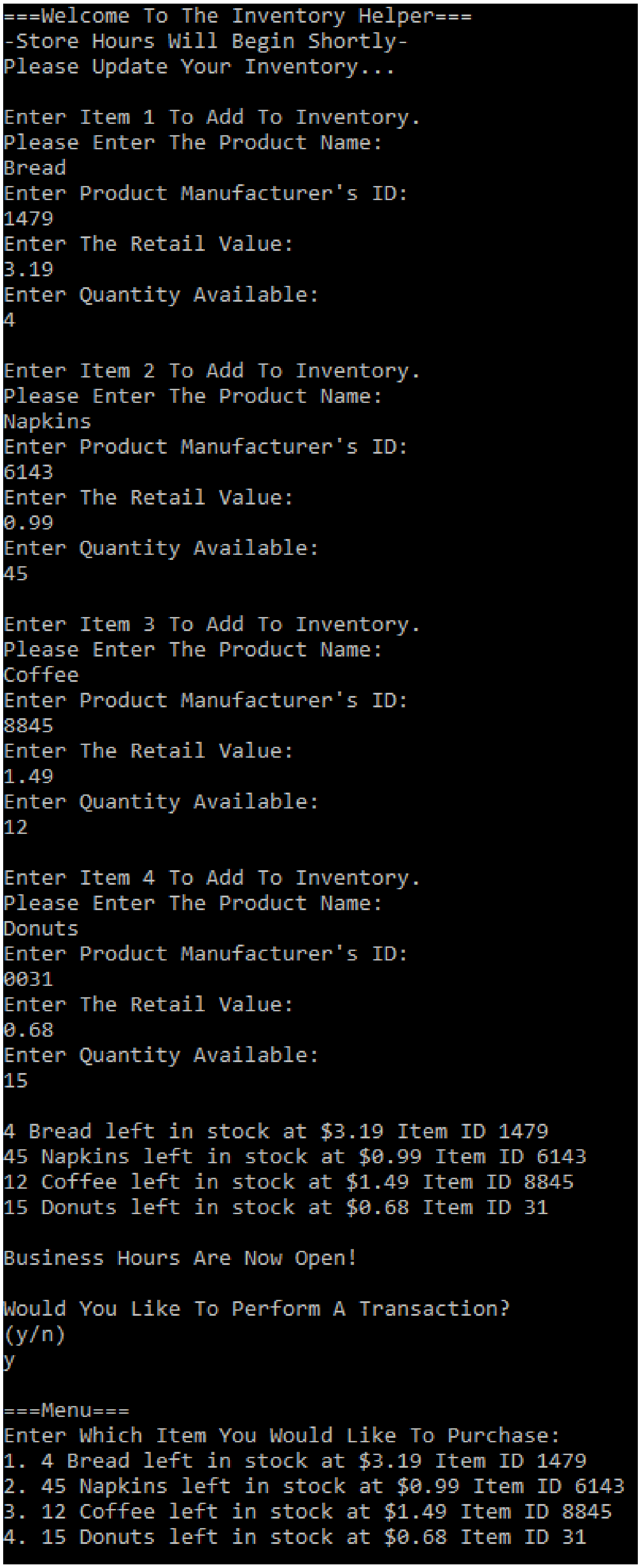
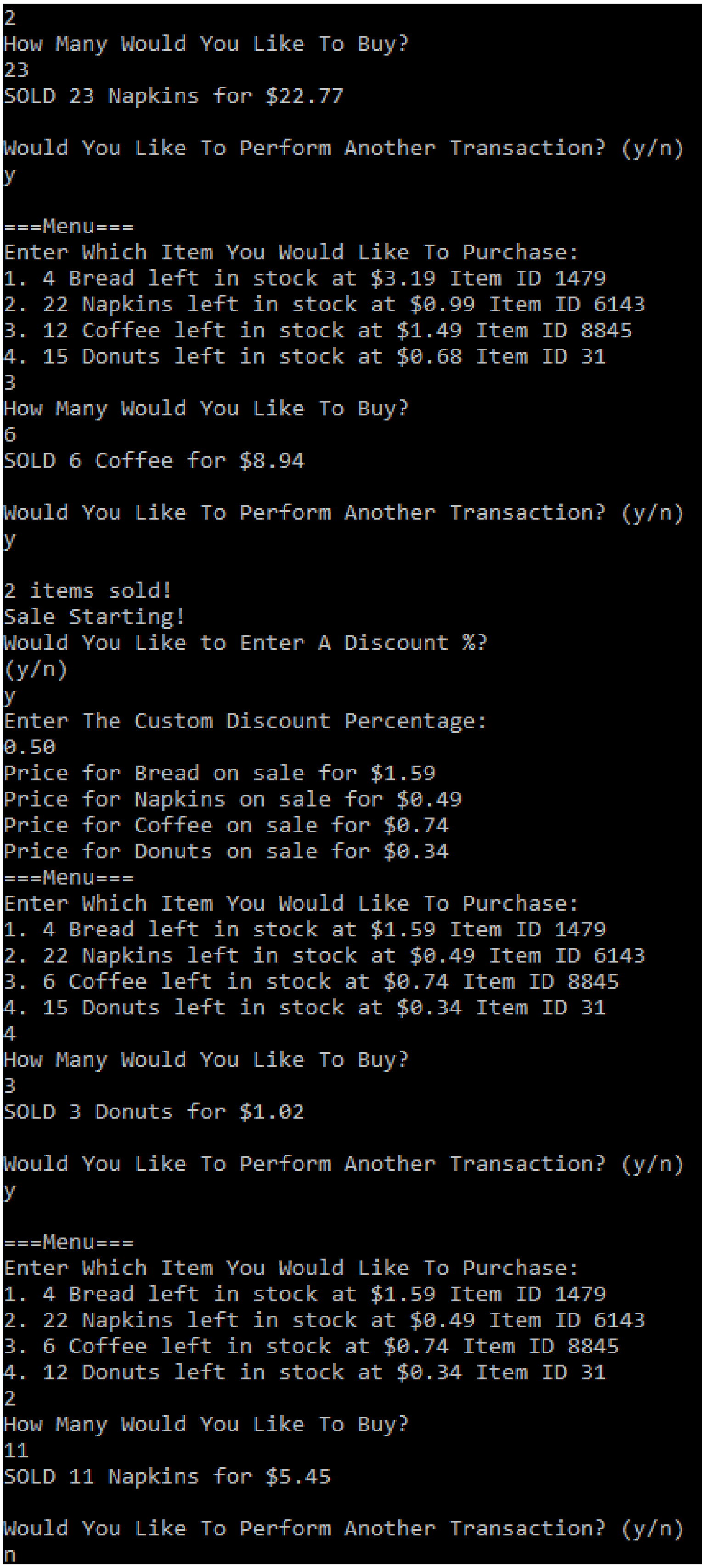
```cpp
/*
Daniel Avila    April 8th, 2020 Section 19
Assignment 2: Inventory Management System
Description: In this assignment using a class and
Description: implementing an array for its objects
*/
#include "Item.h"
const int NUM_ITEMS = 4;//Number of items that are being added to inventory
int main()
{
    Item Inventory[NUM_ITEMS];//Array of 4 Objects
    double discount;//Discount used in new retail values
    char choice, discountChoice;//For the Y's and N's
    int menuChoice, menuQuantity, counter = 0;//For selecting which item to buy
                                                  //how many to buy, and an accumulator
    cout << "===Welcome To The Inventory Helper===" << endl;
    cout << "-Store Hours Will Begin Shortly-" << endl;//The greeting of the program
    cout << "Please Update Your Inventory..." << endl << endl;

    for (int x = 0; x < NUM_ITEMS; x++)
    {
        Inventory[x].addInventory(x);//Adds the items to the Inventory array
    }
    for (int x = 0; x < NUM_ITEMS; x++)
    {
        Inventory[x].displayInventory();//Displays the items from the Inventory array
    }
    cout << endl << "Business Hours Are Now Open!" << endl << endl;
    cout << "Would You Like To Perform A Transaction?" << endl;
    cout << "(y/n)" << endl;
    cin >> choice;//Receives user's transaction choice to follow code for purchase
    cout << endl;
    if (choice == 'y')//When the user chooses 'y', it will make the user select which item,
    {                 //how many of that item and the cost for that quantity of items
        do
        {
            cout << "===Menu===" << endl;
            cout << "Enter Which Item You Would Like To Purchase:" << endl;
            for (int x = 0; x < NUM_ITEMS; x++)
            {
                cout << (x + 1) << ". ";
                Inventory[x].displayInventory();//Displays the inventory and the number
            }                                     //it is for purchase to the user
            cin >> menuChoice;//User's item number
            while (menuChoice < 1 || menuChoice > 4)//Checks to make sure the choice is
            {                                           //between 1 and 4
                cout << "Invalid Option! Try Again:" << endl;
                cin >> menuChoice;//Validates the input until it's within the bounds
            }
            cout << "How Many Would You Like To Buy?" << endl;
            cin >> menuQuantity;//Amount of items being bought
            while (menuQuantity > Inventory[menuChoice - 1].getQuantity())//Checks if the amount is greater
            {//than the amount the user input for that specfic item; it validates the input until it's less than it
                cout << "Not Enough " << Inventory[menuChoice - 1].getProductName() << " In Stock ("
                    << Inventory[menuChoice - 1].getQuantity() << ") Total Stock!" << endl;
                cout << "Enter New Amount:" << endl;
                cin >> menuQuantity;//Keeps validating until it's within bounds
            }//Using [menuChoice - 1] to compensate for the user's item in the correct array index(1,2,3,4) ->(0,1,2,3)
            cout << "SOLD " << menuQuantity << " " << Inventory[menuChoice - 1].getProductName();
            cout << " for $" << (Inventory[menuChoice - 1].getRetailValue() *menuQuantity) << endl << endl;//Calculates total cost
            Inventory[menuChoice - 1].setNewQuantity(menuQuantity);//Subtracts user quantity from overal quantity

            cout << "Would You Like To Perform Another Transaction? (y/n)" << endl;
            cin >> choice;
            counter++;//Accumulator is added by one if the response is 'y' because it will reach here again to add only when it is 'y'
            cout << endl;

            if (counter == 2)//Once the user has input two 'y' then discount is applied to all the retail values and new values are displayed for them
            {
                cout << counter << " items sold!" << endl; //2 items sold
                cout << "Sale Starting!" << endl;
                cout << "Would You Like to Enter A Discount %?" << endl;
                cout << "(y/n)" << endl;
                cin >> discountChoice;
                if (discountChoice == 'y')//When the choice is 'y' then the user inputs a unique discount percentage for all costs
                {
                    cout << "Enter The Custom Discount Percentage:" << endl;
                    cin >> discount;//The discount amount
                    for (int x = 0; x < NUM_ITEMS; x++)
                    {
                        Inventory[x].setDiscountedRV(discount);//Sets the
                    discount percentage to all the retail values
                    }
                    for (int x = 0; x < NUM_ITEMS; x++)//Displays the new values with the discount applies
                    {
                        cout << "Price for ";
                        cout << Inventory[x].getProductName() << " ";
                        cout << "on sale for $";
                        cout << Inventory[x].getRetailValue();
                        cout << endl;
                    }
                }
                else if (discountChoice == 'n')//If the discount choice is 'n' then nothing occurs
                {
                    cout << endl;
                }
            }
        } while (choice != 'n');//Repeats as long as the answer is 'y' or when it is 'n', it ends
    }
    if (choice == 'n')//When the choice is 'n', the final display is shown
    {
        cout << "Closing Shop -- Inventory Left" << endl;
        for (int x = 0; x < NUM_ITEMS; x++)
        {
            Inventory[x].displayInventory();//Shows the remaining inventory with the discounted prices
        }
    }
    system("pause>nul");
    return 0;
}
```

```cpp
#ifndef ITEM_H
#define ITEM_H
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

class Item
{
private:
    string productName;//Name
    int manufacturerID;//ID
    int quantity;//Amount
    double retailValue;//Cost
public:
    Item();//Default Constructor

    void setNewQuantity(int menuQuantity);//User Quantity - Overall Quantity
    void setDiscountedRV(double discount);//Discount to Retail Value

    string getProductName();
    int getManufacturerID();
    int getQuantity();
    double getRetailValue();

    void addInventory(int cnt);//Inititializes Inventory from User
    void displayInventory();//Prints Inventory by User
};

Item::Item()
{//Default values for the privates
    productName = " ";
    manufacturerID = 0;
    quantity = 0;
    retailValue = 0.0;
}
void Item::setNewQuantity(int menuQuantity)
{
    quantity = quantity - menuQuantity;//New quantity to display
}
void Item::setDiscountedRV(double discount)
{
    retailValue = retailValue - (retailValue*discount);//New cost for product
}
string Item::getProductName()
{
    return productName;
}
int Item::getManufacturerID()
{
    return manufacturerID;
}
```

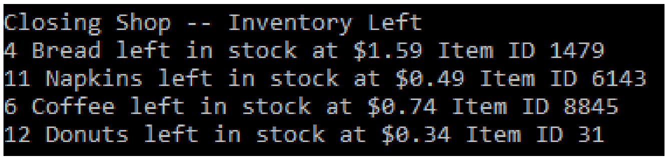
```cpp
int Item::getQuantity()
{
    return quantity;
}
double Item::getRetailValue()
{
    return retailValue;
}
void Item::addInventory(int cnt)
{
    cout << "Enter Item " << (cnt + 1) << " To Add To Inventory." << endl;
    cout << "Please Enter The Product Name:" << endl;
    getline(cin, productName);//Takes product name
    cout << "Enter Product Manufacturer's ID:" << endl;
    cin >> manufacturerID;//takes ID
    cout << "Enter The Retail Value:" << endl;
    cin >> retailValue;//takes cost
    cout << "Enter Quantity Available:" << endl;
    cin >> quantity;//takes amount
    cin.ignore();//used to forget previous input
    cout << endl;
}
void Item::displayInventory()
{
    cout << getQuantity() << " " << getProductName() << " left in stock at $";
    cout << fixed << setprecision(2) << getRetailValue() << " Item ID ";//for two places after decimal
    cout << getManufacturerID() << endl;
}
#endif // !ITEM_H
```



```
===Welcome To The Inventory Helper===
-Store Hours Will Begin Shortly-
Please Update Your Inventory...

Enter Item 1 To Add To Inventory.
Please Enter The Product Name:
Bread
Enter Product Manufacturer's ID:
1479
Enter The Retail Value:
3.19
Enter Quantity Available:
4

Enter Item 2 To Add To Inventory.
Please Enter The Product Name:
Napkins
Enter Product Manufacturer's ID:
6143
Enter The Retail Value:
0.99
Enter Quantity Available:
45

Enter Item 3 To Add To Inventory.
Please Enter The Product Name:
Coffee
Enter Product Manufacturer's ID:
8845
Enter The Retail Value:
1.49
Enter Quantity Available:
12

Enter Item 4 To Add To Inventory.
Please Enter The Product Name:
Donuts
Enter Product Manufacturer's ID:
0031
Enter The Retail Value:
0.68
Enter Quantity Available:
15

4 Bread left in stock at $3.19 Item ID 1479
45 Napkins left in stock at $0.99 Item ID 6143
12 Coffee left in stock at $1.49 Item ID 8845
15 Donuts left in stock at $0.68 Item ID 31

Business Hours Are Now Open!

Would You Like To Perform A Transaction?
(y/n)
y

===Menu===
Enter Which Item You Would Like To Purchase:
1. 4 Bread left in stock at $3.19 Item ID 1479
2. 45 Napkins left in stock at $0.99 Item ID 6143
3. 12 Coffee left in stock at $1.49 Item ID 8845
4. 15 Donuts left in stock at $0.68 Item ID 31
```



```
2
How Many Would You Like To Buy?
23
SOLD 23 Napkins for $22.77

Would You Like To Perform Another Transaction? (y/n)
y

===Menu===
Enter Which Item You Would Like To Purchase:
1. 4 Bread left in stock at $3.19 Item ID 1479
2. 22 Napkins left in stock at $0.99 Item ID 6143
3. 12 Coffee left in stock at $1.49 Item ID 8845
4. 15 Donuts left in stock at $0.68 Item ID 31
3
How Many Would You Like To Buy?
6
SOLD 6 Coffee for $8.94

Would You Like To Perform Another Transaction? (y/n)
y

2 items sold!
Sale Starting!
Would You Like to Enter A Discount %?
(y/n)
y
Enter The Custom Discount Percentage:
0.50
Price for Bread on sale for $1.59
Price for Napkins on sale for $0.49
Price for Coffee on sale for $0.74
Price for Donuts on sale for $0.34
===Menu===
Enter Which Item You Would Like To Purchase:
1. 4 Bread left in stock at $1.59 Item ID 1479
2. 22 Napkins left in stock at $0.49 Item ID 6143
3. 6 Coffee left in stock at $0.74 Item ID 8845
4. 15 Donuts left in stock at $0.34 Item ID 31
4
How Many Would You Like To Buy?
3
SOLD 3 Donuts for $1.02

Would You Like To Perform Another Transaction? (y/n)
y

===Menu===
Enter Which Item You Would Like To Purchase:
1. 4 Bread left in stock at $1.59 Item ID 1479
2. 22 Napkins left in stock at $0.49 Item ID 6143
3. 6 Coffee left in stock at $0.74 Item ID 8845
4. 12 Donuts left in stock at $0.34 Item ID 31
2
How Many Would You Like To Buy?
11
SOLD 11 Napkins for $5.45

Would You Like To Perform Another Transaction? (y/n)
n
```



```
Closing Shop -- Inventory Left
4 Bread left in stock at $1.59 Item ID 1479
11 Napkins left in stock at $0.49 Item ID 6143
6 Coffee left in stock at $0.74 Item ID 8845
12 Donuts left in stock at $0.34 Item ID 31
```