

```
1  /*
2     Zach Hofmeister      3/1/19
3     Assignment 2: Object Oriented Programming - Inventory management system
4     Description: This program keeps track of a store's inventory
5  */
6
7  #include "pch.h"
8  #include <iostream>
9  #include <iomanip>
10 #include "Item.h"
11
12 using namespace std;
13
14 void displayItems(Item[4]); //Displays a list / menu of the items available in
15 void performTransaction(Item[4]); //Allows the user to purchase items from the
16
17 int main() {
18     cout << fixed << showpoint << setprecision(2);
19     cout << "===Welcome to the inventory helper===" << endl;
20     cout << "Store hours will begin shortly." << endl;
21     cout << "Please update your inventory..." << endl;
22     cout << endl;
23
24     Item items[4]; //Inventory array
25     for (int i = 0; i < 4; i++) { //For loop for initializing the inventory of
26         string name;
27         int id, amount;
28         double price;
29
30         cout << "Enter item " << i + 1 << " to add to inventory." << endl;
31         cout << "Please enter the product name: ";
32         cin >> name;
33         cout << "Enter the manufacturer's ID: ";
34         cin >> id;
35         cout << "Enter the retail value: ";
36         cin >> price;
37         cout << "Enter the quantity available: ";
38         cin >> amount;
39         cout << endl;
40
41         items[i] = Item(name, id, price, amount); //Uses overloaded constructor
42         to set the private values for each item.
43     }
44
45     cout << endl;
46     displayItems(items); //Displays the inventory just created.
47     cout << "Business hours are now open!" << endl;
48     performTransaction(items); //Recursive function for transactions.
```

```
49     cout << endl << "Closing shop - inventory left:" << endl;
50     displayItems(items); //Displays the final inventory.
51
52     return 0;
53 }
54
55 void displayItems(Item items[4]) { //Displays a list / menu of the items available in the store. Inventory is passed in to display.
56     for (int i = 0; i < 4; i++) {
57         cout << i+1 << ". " << items[i].getAmount() << " " << items[i].getName() << " left in stock at $" << items[i].getPrice() << " item id " << items[i].getID() << endl;
58     }
59 }
60
61 void performTransaction(Item items[4]) { //Allows the user to purchase items from the inventory. Recursive. Inventory is passed in to access.
62     static int runs = 0; //keeps track of runs to see if a purchase has already been made once.
63     char input;
64     cout << endl;
65     cout << "Would you like to perform " << (runs < 1? "a" : "another") << " transaction? (y/n): " << endl;
66     cin >> input;
67
68     if (input == 'y') { //Make a purchase
69         int itemSelection = 0, amount = 0;
70         cout << "===Menu===" << endl;
71         displayItems(items);
72         do {
73             cout << "Enter an item which you would like to purchase: ";
74             cin >> itemSelection;
75             if (itemSelection < 1 || itemSelection > 4) { //Input validation.
76                 cout << "Invalid item selection. Please enter 1-4." << endl;
77             }
78         } while (itemSelection < 1 || itemSelection > 4); //Input validation.
79
80         do {
81             cout << "How many: ";
82             cin >> amount;
83             if (amount < 1 || amount > items[itemSelection - 1].getAmount()) { //Input validation.
84                 cout << "Invalid amount. Current stock total is " << items[itemSelection - 1].getAmount() << "." << endl;
85             }
86         } while (amount < 1 || amount > items[itemSelection-1].getAmount()); //Input validation.
87         cout << "SOLD " << amount << " " << items[itemSelection-1].getName() << " for $" << items[itemSelection-1].getPrice() * amount << endl;
88         items[itemSelection-1].setAmount(items[itemSelection-1].getAmount() - amount); //Subtracts purchased items from the inventory.
89         runs++;
```

```
190     performTransaction(items); //Recursion
191 } else if (input != 'y' && input != 'n') { //Input validation.
192     cout << "Invalid input. Please try again." << endl;
193     performTransaction(items); //Recursion
194 }
195 }
196
197 /*
198 SAMPLE OUTPUT
199 ===Welcome to the inventory helper===
100 Store hours will begin shortly.
101 Please update your inventory...
102
103 Enter item 1 to add to inventory.
104 Please enter the product name: Milk
105 Enter the manufacturer's ID: 1
106 Enter the retail value: 4.45
107 Enter the quantity available: 10
108
109 Enter item 2 to add to inventory.
110 Please enter the product name: Cookies
111 Enter the manufacturer's ID: 2
112 Enter the retail value: 1.00
113 Enter the quantity available: 40
114
115 Enter item 3 to add to inventory.
116 Please enter the product name: Roses
117 Enter the manufacturer's ID: 3
118 Enter the retail value: 2.00
119 Enter the quantity available: 12
120
121 Enter item 4 to add to inventory.
122 Please enter the product name: Carrots
123 Enter the manufacturer's ID: 4
124 Enter the retail value: .59
125 Enter the quantity available: 33
126
127
128 1. 10 Milk left in stock at $4.45 item id 1
129 2. 40 Cookies left in stock at $1.00 item id 2
130 3. 12 Roses left in stock at $2.00 item id 3
131 4. 33 Carrots left in stock at $0.59 item id 4
132 Business hours are now open!
133
134 Would you like to perform a transaction? (y/n):
135 y
136 ===Menu===
137 1. 10 Milk left in stock at $4.45 item id 1
138 2. 40 Cookies left in stock at $1.00 item id 2
139 3. 12 Roses left in stock at $2.00 item id 3
140 4. 33 Carrots left in stock at $0.59 item id 4
141 Enter an item which you would like to purchase: 2
```

```
142 How many: 50
143 Invalid amount. Current stock total is 40.
144 How many: 25
145 SOLD 25 Cookies for $25.00
146
147 Would you like to perform another transaction? (y/n):
148 y
149 ===Menu===
150 1. 10 Milk left in stock at $4.45 item id 1
151 2. 15 Cookies left in stock at $1.00 item id 2
152 3. 12 Roses left in stock at $2.00 item id 3
153 4. 33 Carrots left in stock at $0.59 item id 4
154 Enter an item which you would like to purchase: 12
155 Invalid item selection. Please enter 1-4.
156 Enter an item which you would like to purchase: 3
157 How many: 12
158 SOLD 12 Roses for $24.00
159
160 Would you like to perform another transaction? (y/n):
161 y
162 ===Menu===
163 1. 10 Milk left in stock at $4.45 item id 1
164 2. 15 Cookies left in stock at $1.00 item id 2
165 3. 0 Roses left in stock at $2.00 item id 3
166 4. 33 Carrots left in stock at $0.59 item id 4
167 Enter an item which you would like to purchase: 4
168 How many: 23
169 SOLD 23 Carrots for $13.57
170
171 Would you like to perform another transaction? (y/n):
172 n
173
174 Closing shop - inventory left:
175 1. 10 Milk left in stock at $4.45 item id 1
176 2. 15 Cookies left in stock at $1.00 item id 2
177 3. 0 Roses left in stock at $2.00 item id 3
178 4. 10 Carrots left in stock at $0.59 item id 4
179
180 Press any key to close this window . . .
181 */
```