

Assignment 1: Item Shop Program**Deadline:** Sunday, September 18 @ 11:59 PMDescription

Your assignment is to write a program called **ItemShop** (in the file **ItemShop.java**) that simulates a simple item shop that allows the user to select an item to purchase, along with the quantity of that item. After obtaining the item selection and desired quantity from the user, your program should process the transaction and display the transaction results to the user.

This code will later be integrated into the game that you will create in Assignment 2.

Program Requirements

Your program must run as follows:

- a. Prompt the user for his/her name. Read the user's input and store it in your program. Do the same for the number of gold coins that the user possesses (read this in as an integer).
- b. Display the following welcome message to the user, where **username** is the name entered by the user:

Welcome to The Item Shop, **username**!

- c. Display the menu on the screen, with each item and the item's cost listed on a numbered line (see the sample runs below for an example to follow). Here are the items and the cost of each item (in units of gold coins):

Item	Cost
Long Sword	120
Short Sword	90
Mace	80
Magic Ring	150
Healing Potion	10

- d. Prompt the user to select an item. The user will select an item by entering the number that corresponds to the desired item. Read the user's input and store it in your program.
- e. Prompt the user to enter the quantity for the selected item. Read the user's input and store it in your program.

Assumption #1: You may assume that the user will enter a valid item number and a valid number for the item quantity for prompted for each of these.

f. Calculate the total cost of the order. **Orders of 3 or more items (i.e. orders where the quantity is at least 3) should receive a 10% discount.** If the user earned a discount, calculate the amount of the discount (in units of gold coins).

Assumption #2: Assume that gold coins are atomic units, and so we can never have a fraction of a gold coin. Therefore, you should always report gold coins in integral units (i.e. as whole numbers, so as 3 and not as 3.0). If a calculation yields a fractional value of gold coins (e.g. 3.5 gold coins), simply truncate the fractional part (e.g. 3.5 becomes 3). However, all prices are a multiple of 10, and so you should never have a situation where the total cost calculations yields a fraction of a coin.

g. Calculate the final cost by subtracting the discount amount from the total.

h. Display the total cost (i.e. cost before discount), discount amount (display “0” if no discount is awarded), and final cost to the user (see the sample runs below for an example to follow).

i. Determine if the user’s transaction can be processed by comparing the final cost to the user’s number of gold coins.

If the user has sufficient funds, then display the following message to the user, where **username** is the name entered by the user:

Thank you, **username**! Your transaction is complete! Please stop by again!

If the user has insufficient funds, then instead display the following:

username - you have insufficient funds! Please come back with more gold!

Coding Requirements

1. Store the price of each item in a **constant** (i.e. a variable initialized with the keyword `final` – see Slide 24 in the Lecture4 slides), and use these constants in all calculations that involve the price of an item instead of using the corresponding numeric literals in the calculations.

2. Include a **multi-line comment** at the top of your source file that includes your name, the class name (CS401), the semester (Fall 2016), and the assignment name (Assignment 1), with each item on its own line.

3. Use good programming style.

Optional: Feel free to make creative changes to the output messages and item names.

Three Example Program Runs

1. Successful purchase with discount (user input is in **bold, red** font)

Enter your name: **Dennis**

Enter your number of gold coins: **550**

Welcome to The Item Shop, Dennis!

Here's what we have for sale (all prices are in units of gold):

- | | |
|-------------------|-----|
| 1. Long Sword | 120 |
| 2. Short Sword | 90 |
| 3. Mace | 80 |
| 4. Magic Ring | 150 |
| 5. Healing Potion | 10 |

Please enter the item number: **2**

Please enter the quantity: **3**

Total cost: 270 gold

Discount: 27 gold

Final cost: 243 gold

Thank you, Dennis! Your transaction is complete! Please stop by again!

2. Successful purchase with no discount (user input is in **bold, red** font)

Enter your name: **Dee**

Enter your number of gold coins: **680**

Welcome to The Item Shop, Dee!

Here's what we have for sale (all prices are in units of gold):

- | | |
|-------------------|-----|
| 1. Long Sword | 120 |
| 2. Short Sword | 90 |
| 3. Mace | 80 |
| 4. Magic Ring | 150 |
| 5. Healing Potion | 10 |

Please enter the item number: **1**

Please enter the quantity: **1**

Total cost: 120 gold

Discount: 0 gold

Final cost: 120 gold

Thank you, Dee! Your transaction is complete! Please stop by again!

3. Unsuccessful purchase without discount (user input is in **bold, red** font)

Enter your name: **Frank**

Enter your number of gold coins: **19**

Welcome to The Item Shop, Frank!

Here's what we have for sale (all prices are in units of gold):

- | | |
|-------------------|-----|
| 1. Long Sword | 120 |
| 2. Short Sword | 90 |
| 3. Mace | 80 |
| 4. Magic Ring | 150 |
| 5. Healing Potion | 10 |

Please enter the item number: **5**

Please enter the quantity: **2**

Total cost: 20 gold

Discount: 0 gold

Final cost: 20 gold

Frank - you have insufficient funds! Please come back with more gold!