

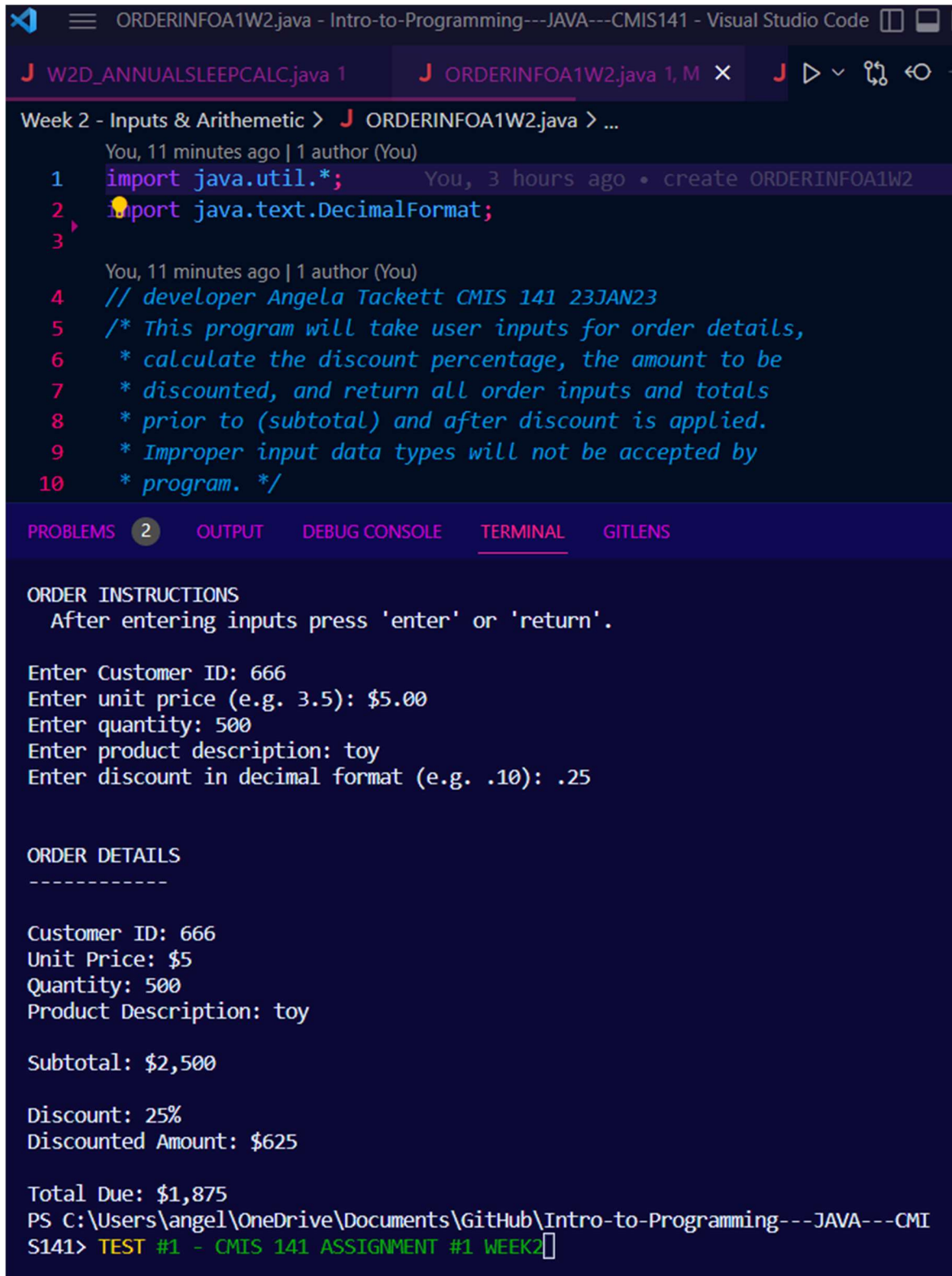
# ASSIGNMENT 1 - WEEK 2 CMIS 141

ANGELA D. TACKETT

23JUN2023

Input	Expected Output	Actual Output	Pass? (Y/N)
customer_id: 666 unit_price: 5.00 quantity: 500 prod_description: toy discount: .25	Customer ID: 666 Unit Price: \$5 Quantity: 500 Product Description: toy Subtotal: \$2,500 Discount: 25% Discounted Amount: \$625 Total Due: \$1,875	Customer ID: 666 Unit Price: \$5 Quantity: 500 Product Description: toy Subtotal: \$2,500 Discount: 25% Discounted Amount: \$625 Total Due: \$1,875	Y
customer_id: ANGELA customer_id: 8547 unit_price: 985.654 quantity: 6254 prod_description: gemstone discount: .60	Customer_id: error Customer ID: 8547 Unit Price: \$985.65 Quantity: 6254 Product Description: gemstone Subtotal: \$6,164,280.12 Discount: 60% Discounted Amount: \$3,698,568.07 Total Due: \$2,465,712.05	Customer_id: error Customer ID: 8547 Unit Price: \$985.65 Quantity: 6254 Product Description: gemstone Subtotal: \$6,164,280.12 Discount: 60% Discounted Amount: \$3,698,568.07 Total Due: \$2,465,712.05	Y
customer_id: 123456 unit_price:45.23 quantity: 15 prod_description: SOAP discount: .72	Customer ID: 123456 Unit Price: \$45.23 Quantity: 15 Product Description: SOAP Subtotal: \$678.45 Discount: 72% Discounted Amount: \$488.48 Total Due: \$189.97	Customer ID: 123456 Unit Price: \$45.23 Quantity: 15 Product Description: SOAP Subtotal: \$678.45 Discount: 72% Discounted Amount: \$488.48 Total Due: \$189.97	Y

# TEST CASE # 1



The screenshot shows a Visual Studio Code window with the title bar "ORDERINFOA1W2.java - Intro-to-Programming---JAVA---CMIS141 - Visual Studio Code". The editor has two tabs: "W2D\_ANNUALSLEEPCALC.java 1" and "ORDERINFOA1W2.java 1, M x". The active tab shows a Java program with the following code:

```
1 import java.util.*;
2 import java.text.DecimalFormat;
3
4 // developer Angela Tackett CMIS 141 23JAN23
5 /* This program will take user inputs for order details,
6  * calculate the discount percentage, the amount to be
7  * discounted, and return all order inputs and totals
8  * prior to (subtotal) and after discount is applied.
9  * Improper input data types will not be accepted by
10  * program. */
```

Below the code editor, the "TERMINAL" tab is active, displaying the program's output:

```
ORDER INSTRUCTIONS
  After entering inputs press 'enter' or 'return'.

Enter Customer ID: 666
Enter unit price (e.g. 3.5): $5.00
Enter quantity: 500
Enter product description: toy
Enter discount in decimal format (e.g. .10): .25

ORDER DETAILS
-----

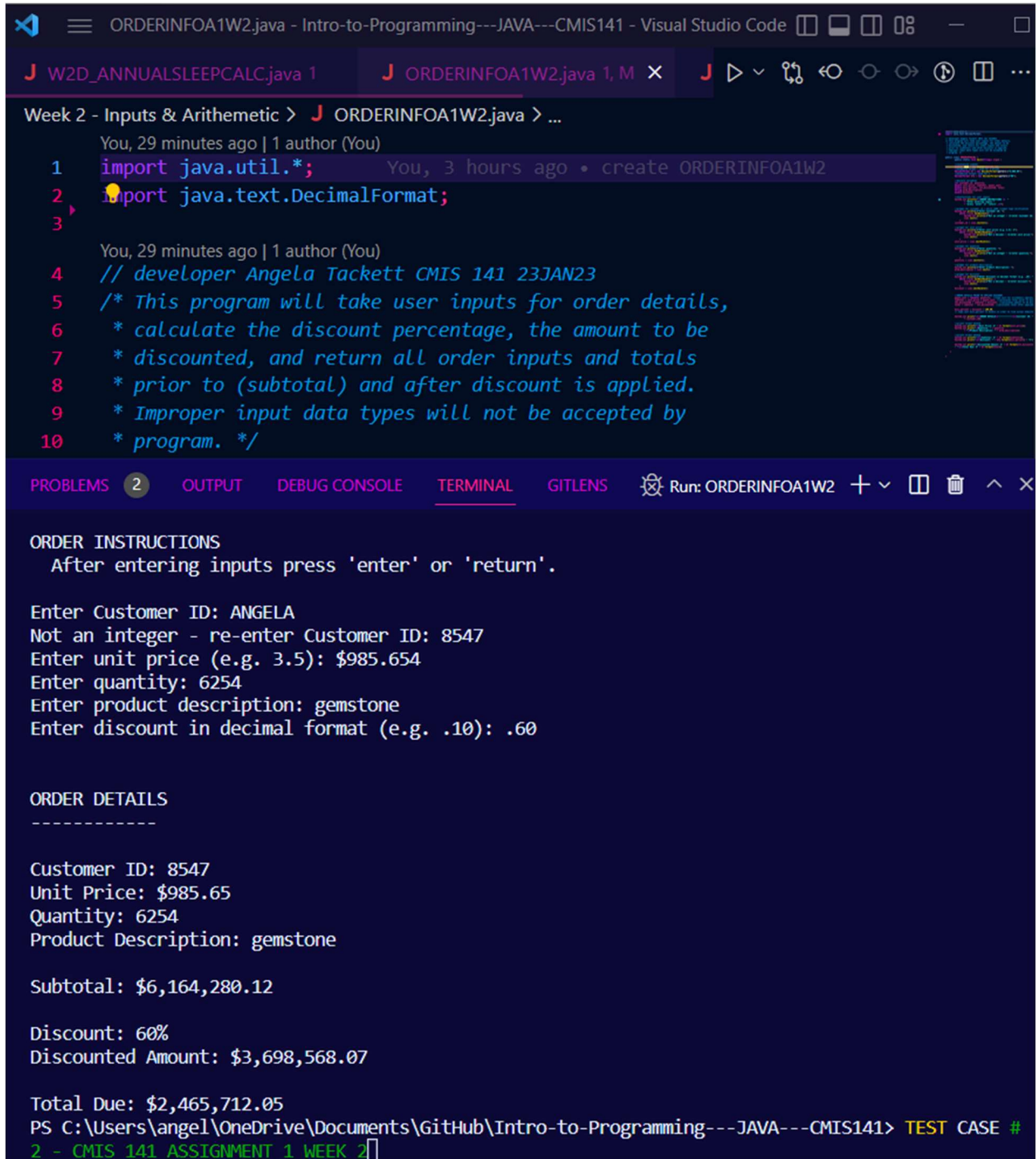
Customer ID: 666
Unit Price: $5
Quantity: 500
Product Description: toy

Subtotal: $2,500

Discount: 25%
Discounted Amount: $625

Total Due: $1,875
PS C:\Users\angel\OneDrive\Documents\GitHub\Intro-to-Programming---JAVA---CMIS141> TEST #1 - CMIS 141 ASSIGNMENT #1 WEEK2
```

# TEST CASE # 2



The screenshot shows a Visual Studio Code window with a Java file named `ORDERINFOA1W2.java`. The code is a Java program that takes user input for order details and calculates the subtotal, discount, and total due. The program is run, and the output is displayed in the terminal.

```
Week 2 - Inputs & Arithmetic > J ORDERINFOA1W2.java > ...
You, 29 minutes ago | 1 author (You)
1 import java.util.*;
2 import java.text.DecimalFormat;
3
4 // developer Angela Tackett CMIS 141 23JAN23
5 /* This program will take user inputs for order details,
6  * calculate the discount percentage, the amount to be
7  * discounted, and return all order inputs and totals
8  * prior to (subtotal) and after discount is applied.
9  * Improper input data types will not be accepted by
10  * program. */
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL GITLENS Run: ORDERINFOA1W2 + -

ORDER INSTRUCTIONS  
After entering inputs press 'enter' or 'return'.

Enter Customer ID: ANGELA  
Not an integer - re-enter Customer ID: 8547  
Enter unit price (e.g. 3.5): \$985.654  
Enter quantity: 6254  
Enter product description: gemstone  
Enter discount in decimal format (e.g. .10): .60

ORDER DETAILS  
-----

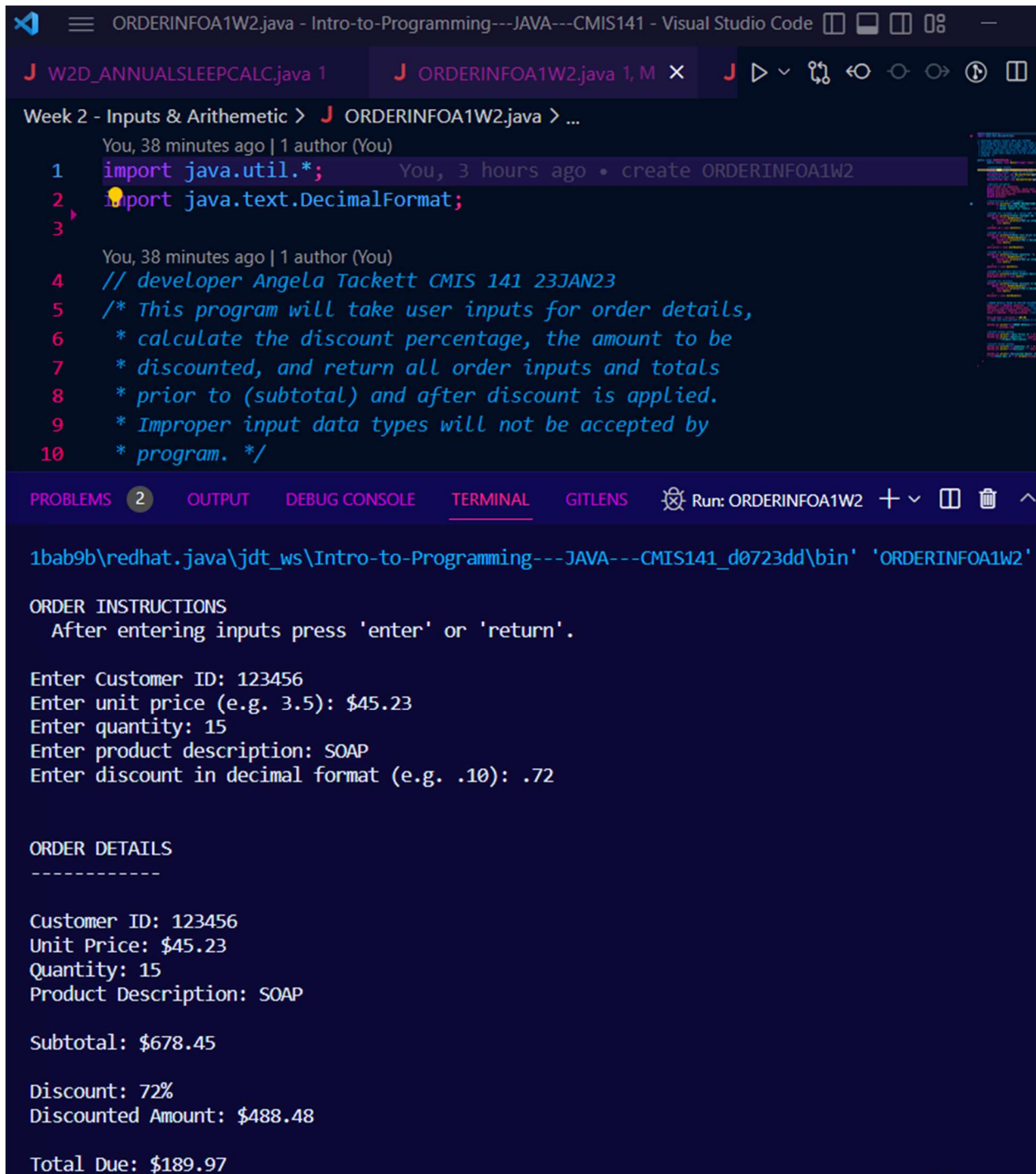
Customer ID: 8547  
Unit Price: \$985.65  
Quantity: 6254  
Product Description: gemstone

Subtotal: \$6,164,280.12

Discount: 60%  
Discounted Amount: \$3,698,568.07

Total Due: \$2,465,712.05  
PS C:\Users\angel\OneDrive\Documents\GitHub\Intro-to-Programming---JAVA---CMIS141> TEST CASE #  
2 - CMIS 141 ASSIGNMENT 1 WEEK 2

# TEST CASE # 3



The screenshot displays the Visual Studio Code interface. The top panel shows the file explorer with 'ORDERINFOA1W2.java' selected. The editor window contains the following Java code:

```
1 import java.util.*;
2 import java.text.DecimalFormat;
3
4 // developer Angela Tackett CMIS 141 23JAN23
5 /* This program will take user inputs for order details,
6  * calculate the discount percentage, the amount to be
7  * discounted, and return all order inputs and totals
8  * prior to (subtotal) and after discount is applied.
9  * Improper input data types will not be accepted by
10  * program. */
```

The bottom panel shows the terminal output for the program 'ORDERINFOA1W2'. The output includes instructions for user input and the resulting order details.

```
1bab9b\redhat.java\jdt_ws\Intro-to-Programming---JAVA---CMIS141_d0723dd\bin' 'ORDERINFOA1W2'

ORDER INSTRUCTIONS
  After entering inputs press 'enter' or 'return'.

Enter Customer ID: 123456
Enter unit price (e.g. 3.5): $45.23
Enter quantity: 15
Enter product description: SOAP
Enter discount in decimal format (e.g. .10): .72

ORDER DETAILS
-----

Customer ID: 123456
Unit Price: $45.23
Quantity: 15
Product Description: SOAP

Subtotal: $678.45

Discount: 72%
Discounted Amount: $488.48

Total Due: $189.97
```



# SCREENSHOTS OF CODE



```
ORDERINFOA1W2.java - Intro-to-Programming---JAVA---CMIS141 - Visual Studio Code
EPCALC.java 1  ORDERINFOA1W2.java 2, M  WEEK2EXERCISES.java 1
Week 2 - Inputs & Arithmetic > ORDERINFOA1W2.java > ORDERINFOA1W2 > main(String[])
You, 10 minutes ago | 1 author (You)
1  import java.util.*;
2  import java.text.DecimalFormat;
3  import java.math.RoundingMode;
4
You, 10 minutes ago | 1 author (You)
5  // developer Angela Tackett CMIS 141 23JAN23
6  /* This program will take user inputs for order details,
7   * calculate the discount percentage, the amount to be
8   * discounted, and return all order inputs and totals
9   * prior to (subtotal) and after discount is applied.
10  * Improper input data types will not be accepted by
11  * program. */
12
13  public class ORDERINFOA1W2 {
14      Run | Debug
15      public static void main(String[] args) {
16
17          //initialize scanner
18          Scanner scan = new Scanner(System.in);
19          //establish format - used for $ prices
20          DecimalFormat df = new DecimalFormat(pattern: "#,###.##");
21          //establish format- used for %
22          DecimalFormat dfp = new DecimalFormat(pattern: "##");
23
24          //declare variables
25          int customer_id, quantity;
26          double unit_price, subtotal, quant_cast;
27          double disc_percent, tot_discounted, total;
28          String prod_description;
29          double discount;
30
31          //Instructions for user inputs
32          System.out.println("ORDER INSTRUCTIONS \n "
33              + "After entering inputs "
34              + "press 'enter' or 'return'.\n");
```

```
34
35 //prompt for customer id + while LOOP ->input type verification
36 System.out.print(s: "Enter Customer ID: ");
37 while (!scan.hasNextInt()) {
38     System.out.print(s: "Not an integer - re-enter Customer ID:");
39     scan.next();
40 }
41 customer_id = scan.nextInt();
42
43 //prompt for unit price
44 System.out.print(s: "Enter unit price (e.g. 3.5): $");
45 while (!scan.hasNextDouble()) {
46     System.out.print(s: "Not a decimal - re-enter unit price:");
47     scan.next();
48 }
49 unit_price = scan.nextDouble();
50
51 //prompt for quantity
52 System.out.print(s: "Enter quantity: ");
53 while (!scan.hasNextInt()) {
54     System.out.print(s: "Not an integer - re-enter quantity:");
55     scan.next();
56 }
57 quantity = scan.nextInt();
58
59 //prompt for product description
60 System.out.print(s: "Enter product description: ");
61 prod_description = scan.next();
62
63 //prompt for discount
64 System.out.print(s: "Enter discount in decimal format (e.g. .10): ");
65 while (!scan.hasNextDouble()) {
66     System.out.print(s: "Not a decimal - re-enter discount:");
67     scan.next();
68 }
69 discount = scan.nextDouble();
70
```

```

71
72 //ORDER DETAILS PRIOR TO APPLIED DISCOUNT
73 quant_cast = (double) quantity; //type cast for arithmetic operation
74 subtotal = unit_price * quant_cast; //calculates total prior to discount added
75 tot_discounted = subtotal * discount; //calculates amount to be discounted
76 total = subtotal - tot_discounted; //calculates total after applied discount
77
78 disc_percent = discount * 100.00;
79 // type cast disc_percent to double in order to find actual deducted
80
81 System.out.print("\n\nORDER DETAILS\n-----\n\nCustomer ID: "
82               + customer_id);
83
84 //printf format method
85 System.out.printf("\nUnit Price: $" + df.format(unit_price));
86 System.out.print("\nQuantity: " + quantity
87               + "\nProduct Description: " + prod_description);
88
89 //printf format method
90 System.out.printf("\n\nSubtotal: $" + df.format(subtotal));
91 System.out.print("\n\nDiscount: " + dfp.format(disc_percent) + "%");
92
93 System.out.print("\nDiscounted Amount: $" + df.format(tot_discounted)
94               + "\n\nTotal Due: $" + df.format(total));
95
96 }
97
98 }
99

```