Project :

Tejaswi Narne

192372120

**Step 1: Open the inventory program**

Step-by-Step Implementation

Make sure you have your project setup from the previous sections.

**Step 2: User input for number of products**

Create a variable named maxSize and prompt the user to enter the number of products they wish to add.

java

Copy code

import java.util.Scanner;

public class Inventory {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int maxSize = -1;

do {

try {

System.out.println("Enter the number of products you would like to add");

System.out.println("Enter 0 (zero) if you do not wish to add products");

maxSize = scanner.nextInt();

if (maxSize < 0) {

System.out.println("Incorrect Value entered");

}

} catch (Exception e) {

System.out.println("Incorrect data type entered!");

scanner.next(); // Clear the input buffer

}

} while (maxSize < 0);

if (maxSize == 0) {

System.out.println("No products required!");

} else {

Product[] products = new Product[maxSize];

for (int i = 0; i < maxSize; i++) {

scanner.nextLine(); // Clear the input buffer

System.out.println("Enter product name:");

String name = scanner.nextLine();

System.out.println("Enter quantity:");

int quantity = scanner.nextInt();

System.out.println("Enter price:");

double price = scanner.nextDouble();

System.out.println("Enter item number:");

int itemNumber = scanner.nextInt();

products[i] = new Product(name, quantity, price, itemNumber);

}

for (Product product : products) {

System.out.println(product);

}

}

scanner.close();

}

}

**Step 3: Error handling for incorrect data types**

Ensure to handle cases where the user inputs a non-integer value.

java

Copy code

// Already included in Step 2 above

Step 4: Modify the ProductTester class to handle multiple products

Create a single-dimensional array to store products if the user inputs a number greater than zero.

java

Copy code

// Included in Step 2 above

Step 5: Populate the array with user input

Get the values for each product field from the user.

java

Copy code

// Included in Step 2 above

Step 6: Display the information for each product

Use a for-each loop to display the details of each product in the array.

java

Copy code

// Included in Step 2 above

Step 7: Clean up unnecessary code

Remove any code that’s not used in this exercise.

java

Copy code

// Ensure to review the code and clean up if needed

Step 8: Save your project

Make sure to save your project after implementing the changes.

Product Class

Here is a simple Product class for reference:

java

Copy code

public class Product {

private String name;

private int quantity;

private double price;

private int itemNumber;

public Product(String name, int quantity, double price, int itemNumber) {

this.name = name;

this.quantity = quantity;

this.price = price;

this.itemNumber = itemNumber;

}

@Override

public String toString() {

return "Product [name=" + name + ", quantity=" + quantity + ", price=" + price + ", itemNumber=" + itemNumber + "]";

}

}

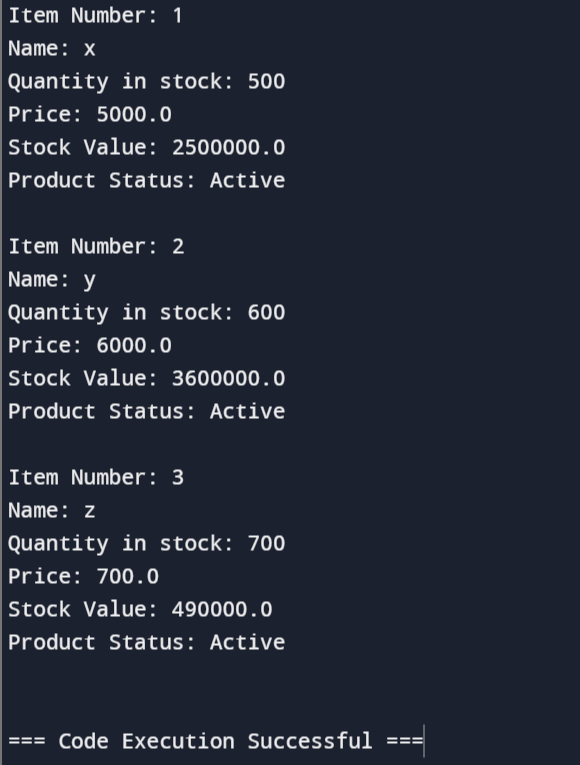
A screenshot of a computer

Description automatically generatedInput:

Output:

A screenshot of a computer

Description automatically generated



A screenshot of a computer program

Description automatically generated