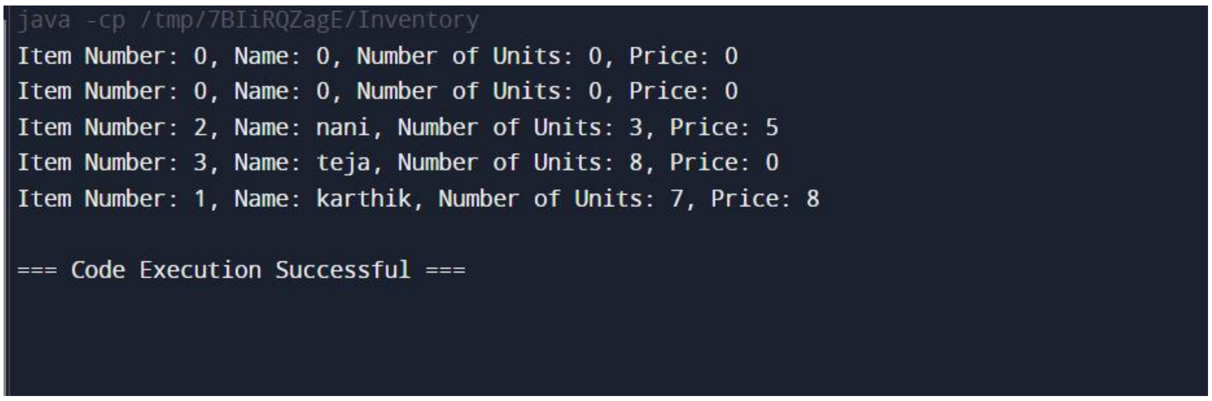
1.project:

```java
public class Inventory {
   private int itemNumber;
   private String name;
   private int numberOfUnits;
   private int price;

   // Default constructor
   public Inventory() {
      this.itemNumber = 0;
      this.name = "0";
      this.numberOfUnits = 0;
      this.price = 0;
   }

   // Parameterized constructor
   public Inventory(int itemNumber, String name, int numberOfUnits, int price) {
      this.itemNumber = itemNumber;
      this.name = name;
      this.numberOfUnits = numberOfUnits;
      this.price = price;
   }

   // Method to set values
   public void set(int itemNumber, String name, int numberOfUnits, int price) {
      this.itemNumber = itemNumber;
      this.name = name;
      this.numberOfUnits = numberOfUnits;
      this.price = price;
   }
```

```
    // Getter methods
    public int getItemNumber() {
        return itemNumber;
    }

    public String getName() {
        return name;
    }

    public int getNumberOfUnits() {
        return numberOfUnits;
    }

    public int getPrice() {
        return price;
    }

    // toString method to print object details
    @Override
    public String toString() {
        return "Item Number: " + itemNumber + ", Name: " + name + ", Number of Units: " +
numberOfUnits + ", Price: " + price;
    }

    // Main method to test the Inventory class
    public static void main(String[] args) {
        Inventory item1 = new Inventory();
        Inventory item2 = new Inventory();
        Inventory item3 = new Inventory(2, "nani", 3, 5);
        Inventory item4 = new Inventory(3, "teja", 8, 0);
        Inventory item5 = new Inventory(1, "karthik", 7, 8);
        System.out.println(item1);
        System.out.println(item2);
        System.out.println(item3);
        System.out.println(item4);
        System.out.println(item5);
    }
}
```

Output:



```
java -cp /tmp/7BIiRQZagE/Inventory
Item Number: 0, Name: 0, Number of Units: 0, Price: 0
Item Number: 0, Name: 0, Number of Units: 0, Price: 0
Item Number: 2, Name: nani, Number of Units: 3, Price: 5
Item Number: 3, Name: teja, Number of Units: 8, Price: 0
Item Number: 1, Name: karthik, Number of Units: 7, Price: 8

=== Code Execution Successful ===
```

2.

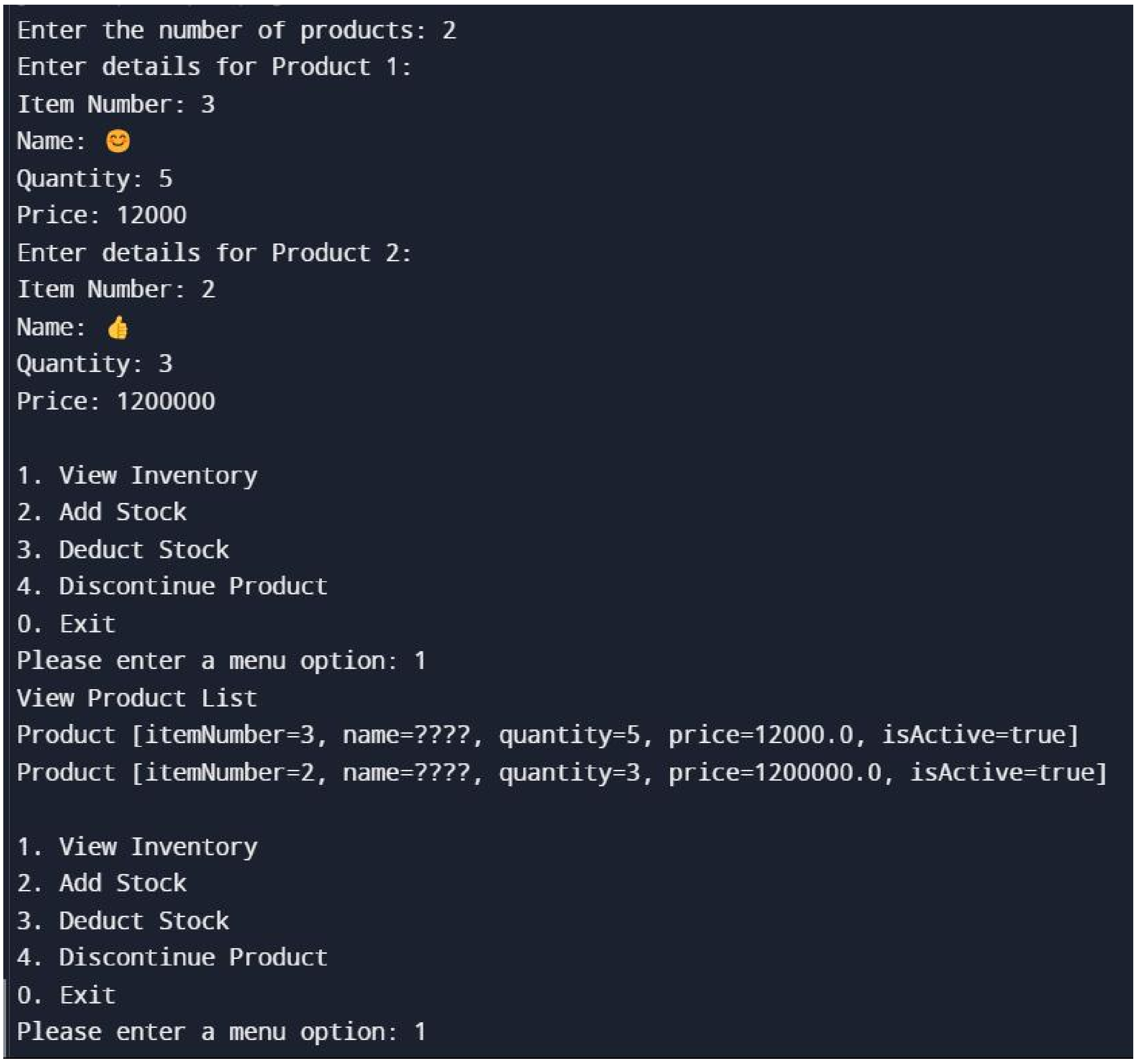
```
import java.util.Scanner;

public class ProductTester {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int maxSize = getNumProducts(in); // Get the number of products
        Product[] products = new Product[maxSize]; // Array to hold products

        addToInventory(products, in); // Add products to the inventory

        int menuChoice;
        do {
            menuChoice = getMenuOption(in); // Display menu and get user's choice
            executeMenuChoice(menuChoice, products, in); // Execute the chosen menu option
        } while (menuChoice != 0); // Exit on choice 0

        in.close();
    }

    public static void displayInventory(Product[] products) {
        for (Product product : products) {
            if (product != null) {
                System.out.println(product);
            }
        }
    }

    public static void addToInventory(Product[] products, Scanner in) {
        for (int i = 0; i < products.length; i++) {
            System.out.println("Enter details for Product " + (i + 1) + ":");
            System.out.print("Item Number: ");
            int tempNumber = in.nextInt();
            in.nextLine(); // Clear the buffer
            System.out.print("Name: ");
            String tempName = in.nextLine();
            System.out.print("Quantity: ");
            int tempQty = in.nextInt();
            System.out.print("Price: ");
            double tempPrice = in.nextDouble();

            products[i] = new Product(tempNumber, tempName, tempQty, tempPrice);
        }
    }
```

```
public static int getNumProducts(Scanner in) {
    System.out.print("Enter the number of products: ");
    return in.nextInt();
}

public static int getMenuOption(Scanner in) {
    int menuChoice = -1;
    while (menuChoice < 0 || menuChoice > 4) {
        System.out.println("\n1. View Inventory");
        System.out.println("2. Add Stock");
        System.out.println("3. Deduct Stock");
        System.out.println("4. Discontinue Product");
        System.out.println("0. Exit");
        System.out.print("Please enter a menu option: ");

        if (in.hasNextInt()) {
            menuChoice = in.nextInt();
        } else {
            in.next(); // Clear the invalid input
        }

        if (menuChoice < 0 || menuChoice > 4) {
            System.out.println("Invalid option. Please try again.");
        }
    }
    return menuChoice;
}

public static int getProductNumber(Product[] products, Scanner in) {
    int productChoice = -1;
```

```
        while (productChoice < 0 || productChoice >= products.length || products[productChoice] ==
null) {
            for (int i = 0; i < products.length; i++) {
                if (products[i] != null) {
                    System.out.println(i + ": " + products[i].getName());
                }
            }
            System.out.print("Please select a product by its number: ");

            if (in.hasNextInt()) {
                productChoice = in.nextInt();
            } else {
                in.next(); // Clear invalid input
            }

            if (productChoice < 0 || productChoice >= products.length || products[productChoice] == null)
{
                System.out.println("Invalid product number. Please try again.");
            }
        }
        return productChoice;
    }

    public static void addInventory(Product[] products, Scanner in) {
        int productChoice = getProductNumber(products, in);
        System.out.print("How many products do you want to add? ");
        int updateValue = -1;

        while (updateValue < 0) {
            if (in.hasNextInt()) {
                updateValue = in.nextInt();
            } else {
```

```
            in.next(); // Clear invalid input
          }

          if (updateValue < 0) {
            System.out.println("Invalid value. Please enter a positive number.");
          }
        }

        products[productChoice].addToInventory(updateValue);
        System.out.println("Updated stock for " + products[productChoice].getName());
    }

    public static void deductInventory(Product[] products, Scanner in) {
        int productChoice = getProductNumber(products, in);
        System.out.print("How many products do you want to deduct? ");
        int updateValue = -1;

        while (updateValue < 0 || updateValue > products[productChoice].getQuantity()) {
          if (in.hasNextInt()) {
            updateValue = in.nextInt();
          } else {
            in.next(); // Clear invalid input
          }

          if (updateValue < 0 || updateValue > products[productChoice].getQuantity()) {
            System.out.println("Invalid value. Please enter a number between 0 and " +
products[productChoice].getQuantity() + ".");
          }
        }

        products[productChoice].deductFromInventory(updateValue);
    System.out.println("Updated stock for " + products[productChoice].getName());
}

public static void discontinueInventory(Product[] products, Scanner in) {
    int productChoice = getProductNumber(products, in);
    products[productChoice].setActive(false);
    System.out.println(products[productChoice].getName() + " has been discontinued.");
}

public static void executeMenuChoice(int menuChoice, Product[] products, Scanner in) {
    switch (menuChoice) {
        case 1:
            System.out.println("View Product List");
            displayInventory(products);
            break;
        case 2:
            System.out.println("Add Stock");
            addInventory(products, in);
            break;
        case 3:
            System.out.println("Deduct Stock");
            deductInventory(products, in);
            break;
        case 4:
            System.out.println("Discontinue Stock");
            discontinueInventory(products, in);
            break;
        case 0:
            System.out.println("Exiting program.");
            break;
        default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}

public class Product {
    // Fields, constructor, and methods from your previous code...

    // Method to add to inventory
    public void addToInventory(int quantity) {
        this.quantity += quantity;
    }

    // Method to deduct from inventory
    public void deductFromInventory(int quantity) {
        this.quantity -= quantity;
    }
}
```

Output:



```
Enter the number of products: 2
Enter details for Product 1:
Item Number: 3
Name: 😊
Quantity: 5
Price: 12000
Enter details for Product 2:
Item Number: 2
Name: 👍
Quantity: 3
Price: 1200000

1. View Inventory
2. Add Stock
3. Deduct Stock
4. Discontinue Product
0. Exit
Please enter a menu option: 1
View Product List
Product [itemNumber=3, name=????, quantity=5, price=12000.0, isActive=true]
Product [itemNumber=2, name=????, quantity=3, price=1200000.0, isActive=true]

1. View Inventory
2. Add Stock
3. Deduct Stock
4. Discontinue Product
0. Exit
Please enter a menu option: 1
```

3.

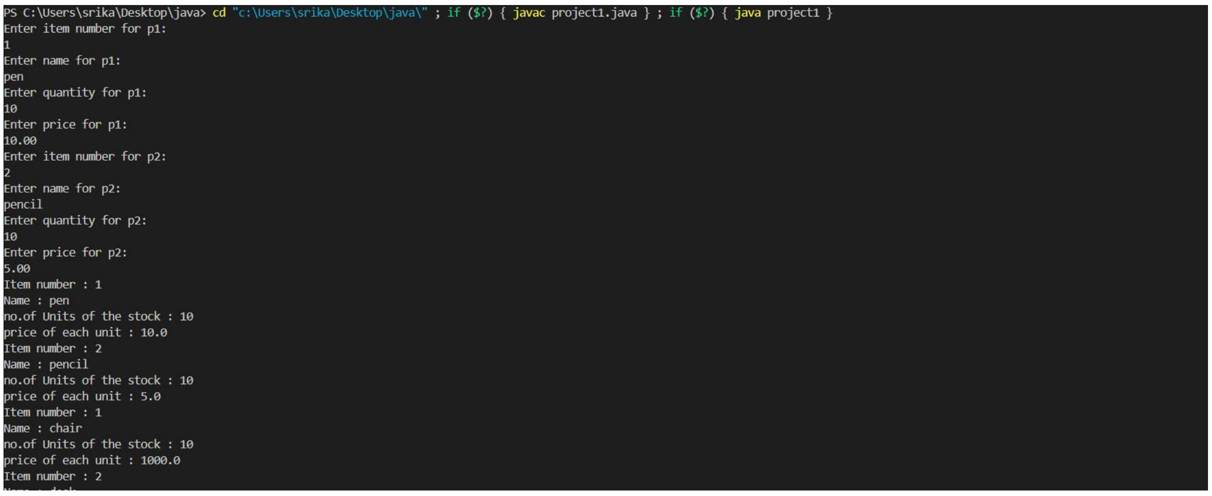
```
import java.util.Scanner;
class Product{
  private int itemNumber;
  private String name;
  private int units;
  private double price;
  public Product(){
    this.itemNumber=0;
    this.name="0";
    this.units=0;
```

```
        this.price=0;

    }

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

        this.itemNumber=itemNumber;

        this.name=name;

        this.units=units;

        this.price=price;

    }

     public void setdata(int itemNumber,String name,int units,double price){

         this.itemNumber=itemNumber;

         this.name=name;

         this.units=units;

        this.price=price;

    }

    public int getnumber(){

        return itemNumber;

    }

    public String getname(){

        return name;

    }

    public int getunits(){

        return units;

    }

    public double getprice(){

        return price;

    }

    public String tostring(){

        return "Item number : "+ itemNumber +"\nName : "+name+"\nno.of Units of the stock :
"+units+"\nprice of each unit : "+price;

    }
```

```
}
public class project1 {
  public static void main(String[] args){
    Scanner in =new Scanner(System.in);
    int tempNumber;
    String tempName;
    int tempQty;
    double tempPrice;
    System.out.println("Enter item number for p1: ");
    tempNumber = in.nextInt();
    in.nextLine();
    System.out.println("Enter name for p1: ");
    tempName = in.nextLine();
    System.out.println("Enter quantity for p1: ");
    tempQty = in.nextInt();
    System.out.println("Enter price for p1: ");
    tempPrice = in.nextDouble();
    Product product1=new Product(tempNumber,tempName,tempQty,tempPrice);
    System.out.println("Enter item number for p2: ");
    tempNumber = in.nextInt();
    in.nextLine();
    System.out.println("Enter name for p2: ");
    tempName = in.nextLine();
    System.out.println("Enter quantity for p2: ");
    tempQty = in.nextInt();
    System.out.println("Enter price for p2: ");
    tempPrice = in.nextDouble();
    Product product2=new Product(tempNumber,tempName,tempQty,tempPrice);
    Product product3=new Product(1,"chair",10,1000.00);
    Product product4=new Product(2,"desk",5,2000.00);
    Product product5=new Product(1,"capboards",3,3000.00);
```

```
        System.out.println(product1.tostring());
        System.out.println(product2.tostring());
        System.out.println(product3.tostring());
        System.out.println(product4.tostring());
        System.out.println(product5.tostring());
        in.close();
    }

}
```

Output:



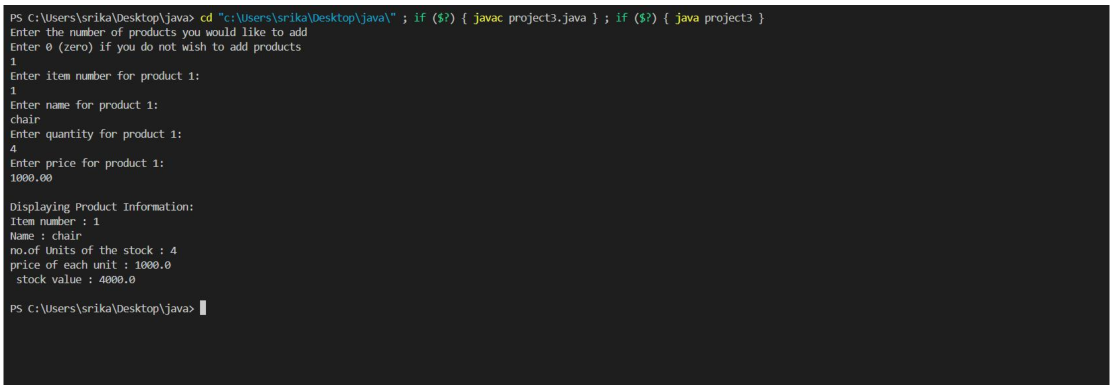
```
PS C:\Users\srika\Desktop\java> cd "c:\Users\srika\Desktop\java\" ; if ($?) { javac project1.java } ; if ($?) { java project1 }
Enter item number for p1:
1
Enter name for p1:
pen
Enter quantity for p1:
10
Enter price for p1:
10.00
Enter item number for p2:
2
Enter name for p2:
pencil
Enter quantity for p2:
10
Enter price for p2:
5.00
Item number : 1
Name : pen
no.of Units of the stock : 10
price of each unit : 10.0
Item number : 2
Name : pencil
no.of Units of the stock : 10
price of each unit : 5.0
Item number : 1
Name : chair
no.of Units of the stock : 10
price of each unit : 1000.0
Item number : 2
```

4.

```
import java.util.Scanner;
class Product{
    private int itemNumber;
    private String name;
    private int units;
    private double price;
    public Product(){
        this.itemNumber=0;
        this.name="0";
        this.units=0;
        this.price=0;
    }
    public Product(int itemNumber,String name,int units,double price){
       this.itemNumber=itemNumber;
       this.name=name;
       this.units=units;
       this.price=price;
    }
     public void setdata(int itemNumber,String name,int units,double price){
        this.itemNumber=itemNumber;
        this.name=name;
        this.units=units;
       this.price=price;
    }
    public int getnumber(){
       return itemNumber;
    }
    public String getname(){
       return name;
    }
    public int getunits(){
       return units;
    }
    public double getprice(){
       return price;
    }
    double inventoryprice(){
       return price*units;
    }
    public String tostring(){
       return "Item number : "+ itemNumber +"\nName : "+name+"\nno.of Units of the stock :
 "+units+"\nprice of each unit : "+price+"\n stock value : "+ inventoryprice();
```

```
    }
}
public class project3 {
    public static void main(String[] args) {
        Scanner in=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 = in.nextInt();
                if (maxSize < 0) {
                    System.out.println("Incorrect value entered. Please enter a positive integer or zero.");
                }
            } catch (Exception e) {
                System.out.println("Incorrect data type entered!");
                in.nextLine();
            }
        } 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++) {
                in.nextLine();
                System.out.println("Enter item number for product " + (i + 1) + ": ");
                int tempNumber = in.nextInt();
                in.nextLine();
                System.out.println("Enter name for product " + (i + 1) + ": ");
                String tempName = in.nextLine();
                System.out.println("Enter quantity for product " + (i + 1) + ": ");
                int tempQty = in.nextInt();
                System.out.println("Enter price for product " + (i + 1) + ": ");
                double tempPrice = in.nextDouble();
                products[i] = new Product(tempNumber, tempName, tempQty, tempPrice);
            }
            System.out.println("\nDisplaying Product Information:");
            for (Product product : products) {
                System.out.println(product.tostring() + "\n");
            }
        }

        in.close();
    }
```

Output:



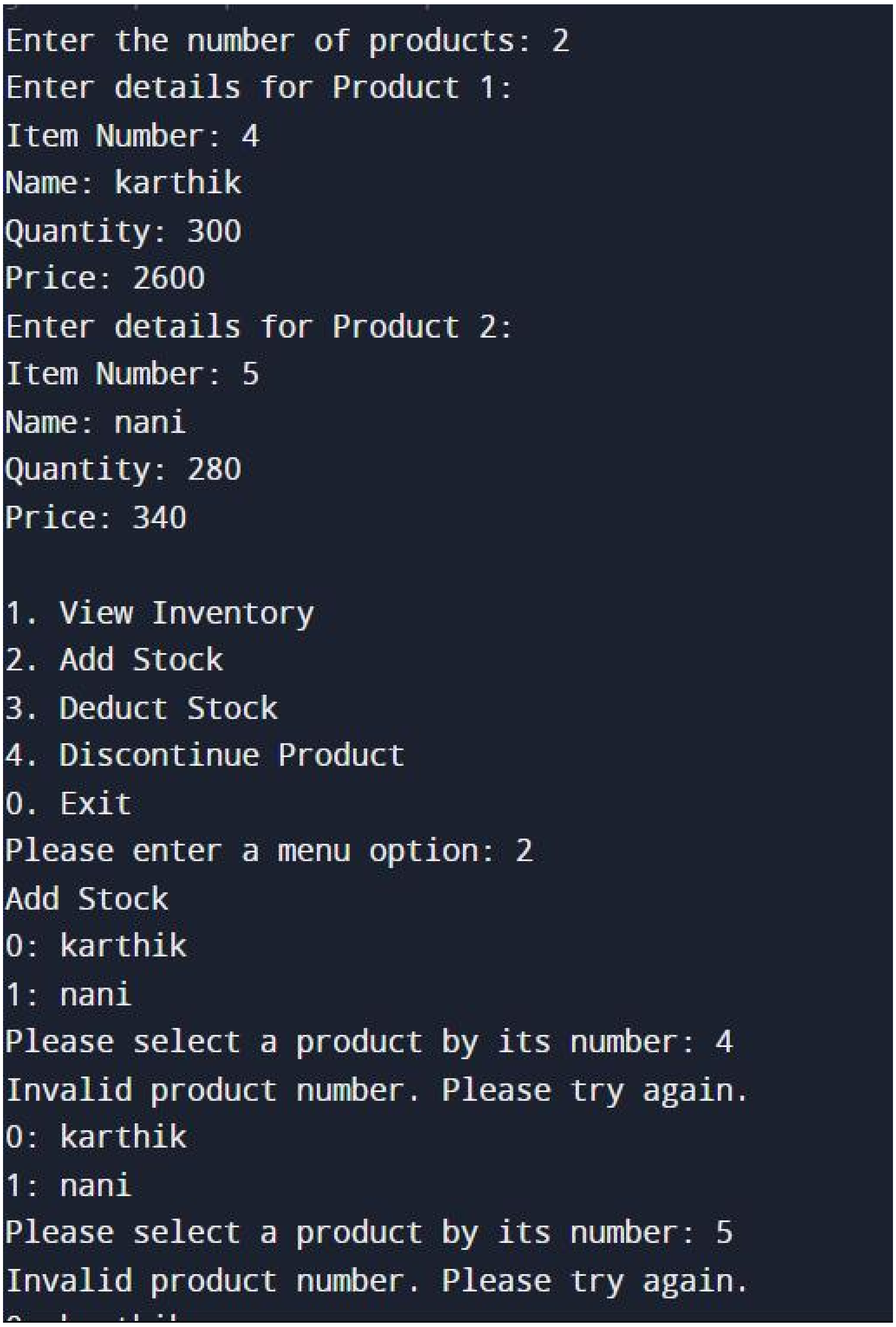
```
PS C:\Users\srika\Desktop\java> cd "c:\Users\srika\Desktop\java\" ; if ($?) { javac project3.java } ; if ($?) { java project3 }
Enter the number of products you would like to add
Enter 0 (zero) if you do not wish to add products
1
Enter item number for product 1:
1
Enter name for product 1:
chair
Enter quantity for product 1:
4
Enter price for product 1:
1000.00

Displaying Product Information:
Item number : 1
Name : chair
no.of Units of the stock : 4
price of each unit : 1000.0
 stock value : 4000.0

PS C:\Users\srika\Desktop\java> 
```

5.

```
import java.util.Scanner;


public class ProductTester {


    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int maxSize = getNumProducts(in);
        Product[] products = new Product[maxSize];
        addToInventory(products, in);

        int menuChoice;
        do {
            menuChoice = getMenuOption(in);
            executeMenuChoice(menuChoice, products, in);
        } while (menuChoice != 0);

        in.close();
    }

    public static void displayInventory(Product[] products) {
        for (Product product : products) {
            if (product != null && product.isActive()) {
                System.out.println(product);
            }
        }
    }

    public static void addToInventory(Product[] products, Scanner in) {
        for (int i = 0; i < products.length; i++) {
            System.out.println("Enter details for Product " + (i + 1) + ":");
            System.out.print("Item Number: ");
            int tempNumber = in.nextInt();
            in.nextLine(); // Clear the buffer
            System.out.print("Name: ");
            String tempName = in.nextLine();
            System.out.print("Quantity: ");
            int tempQty = in.nextInt();
            System.out.print("Price: ");
            double tempPrice = in.nextDouble();
        // Ensure valid inputs
        if (tempQty < 0) tempQty = 0;
        if (tempPrice < 0.0) tempPrice = 0.0;

        products[i] = new Product(tempNumber, tempName, tempQty, tempPrice);
    }
}

public static int getNumProducts(Scanner in) {
    System.out.print("Enter the number of products: ");
    while (!in.hasNextInt()) {
        in.next();
        System.out.print("Invalid input. Enter the number of products: ");
    }
    return in.nextInt();
}

public static int getMenuOption(Scanner in) {
    int menuChoice = -1;
    while (menuChoice < 0 || menuChoice > 4) {
        System.out.println("\n1. View Inventory");
        System.out.println("2. Add Stock");
        System.out.println("3. Deduct Stock");
        System.out.println("4. Discontinue Product");
        System.out.println("0. Exit");
        System.out.print("Please enter a menu option: ");

        if (in.hasNextInt()) {
            menuChoice = in.nextInt();
        } else {
            in.next(); // Clear the invalid input
        }

        if (menuChoice < 0 || menuChoice > 4) {
            System.out.println("Invalid option. Please try again.");
        }
    }
    return menuChoice;
}

public static int getProductNumber(Product[] products, Scanner in) {
    int productChoice = -1;
    while (productChoice < 0 || productChoice >= products.length || products[productChoice] ==
null || !products[productChoice].isActive()) {
        for (int i = 0; i < products.length; i++) {
            if (products[i] != null && products[i].isActive()) {
                System.out.println(i + ": " + products[i].getName());
            }
        }
        System.out.print("Please select a product by its number: ");

        if (in.hasNextInt()) {
            productChoice = in.nextInt();
        } else {
            in.next(); // Clear invalid input
        }

        if (productChoice < 0 || productChoice >= products.length || products[productChoice] == null
|| !products[productChoice].isActive()) {
            System.out.println("Invalid product number. Please try again.");
        }
    }
```

```java
    return productChoice;
}

public static void addInventory(Product[] products, Scanner in) {
    int productChoice = getProductNumber(products, in);
    System.out.print("How many products do you want to add? ");
    int updateValue = -1;

    while (updateValue < 0) {
        if (in.hasNextInt()) {
            updateValue = in.nextInt();
        } else {
            in.next(); // Clear invalid input
        }

        if (updateValue < 0) {
            System.out.println("Invalid value. Please enter a positive number.");
        }
    }

    products[productChoice].addToInventory(updateValue);
    System.out.println("Updated stock for " + products[productChoice].getName());
}

public static void deductInventory(Product[] products, Scanner in) {
    int productChoice = getProductNumber(products, in);
    System.out.print("How many products do you want to deduct? ");
    int updateValue = -1;

    while (updateValue < 0 || updateValue > products[productChoice].getQuantity()) {
        if (in.hasNextInt()) {
                updateValue = in.nextInt();
            } else {
                in.next(); // Clear invalid input
            }

            if (updateValue < 0 || updateValue > products[productChoice].getQuantity()) {
                System.out.println("Invalid value. Please enter a number between 0 and " +
products[productChoice].getQuantity() + ".");
            }
        }

        products[productChoice].deductFromInventory(updateValue);
        System.out.println("Updated stock for " + products[productChoice].getName());
    }

    public static void discontinueInventory(Product[] products, Scanner in) {
        int productChoice = getProductNumber(products, in);
        products[productChoice].setActive(false);
        System.out.println(products[productChoice].getName() + " has been discontinued.");
    }

    public static void executeMenuChoice(int menuChoice, Product[] products, Scanner in) {
        switch (menuChoice) {
            case 1:
                System.out.println("View Product List");
                displayInventory(products);
                break;
            case 2:
                System.out.println("Add Stock");
                addInventory(products, in);
                break;
            case 3:
                System.out.println("Deduct Stock");
                deductInventory(products, in);
                break;
            case 4:
                System.out.println("Discontinue Stock");
                discontinueInventory(products, in);
                break;
            case 0:
                System.out.println("Exiting program.");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}

class Product {
    private int itemNumber;
    private String name;
    private int quantity;
    private double price;
    private boolean active;

    public Product(int itemNumber, String name, int quantity, double price) {
        this.itemNumber = itemNumber;
        this.name = name;
        this.quantity = quantity;
        this.price = price;
        this.active = true;
    }
    public void addToInventory(int quantity) {
        this.quantity += quantity;
    }

    public void deductFromInventory(int quantity) {
        this.quantity -= quantity;
    }

    public void setActive(boolean active) {
        this.active = active;
    }

    public boolean isActive() {
        return active;
    }

    public int getQuantity() {
        return quantity;
    }

    public String getName() {
        return name;
    }

    @Override
    public String toString() {
        return "Product [Item Number=" + itemNumber + ", Name=" + name + ", Quantity=" + quantity +
", Price=" + price + "]";
    }
}
```

Output:



```
Enter the number of products: 2
Enter details for Product 1:
Item Number: 4
Name: karthik
Quantity: 300
Price: 2600
Enter details for Product 2:
Item Number: 5
Name: nani
Quantity: 280
Price: 340

1. View Inventory
2. Add Stock
3. Deduct Stock
4. Discontinue Product
0. Exit
Please enter a menu option: 2
Add Stock
0: karthik
1: nani
Please select a product by its number: 4
Invalid product number. Please try again.
0: karthik
1: nani
Please select a product by its number: 5
Invalid product number. Please try again.
```

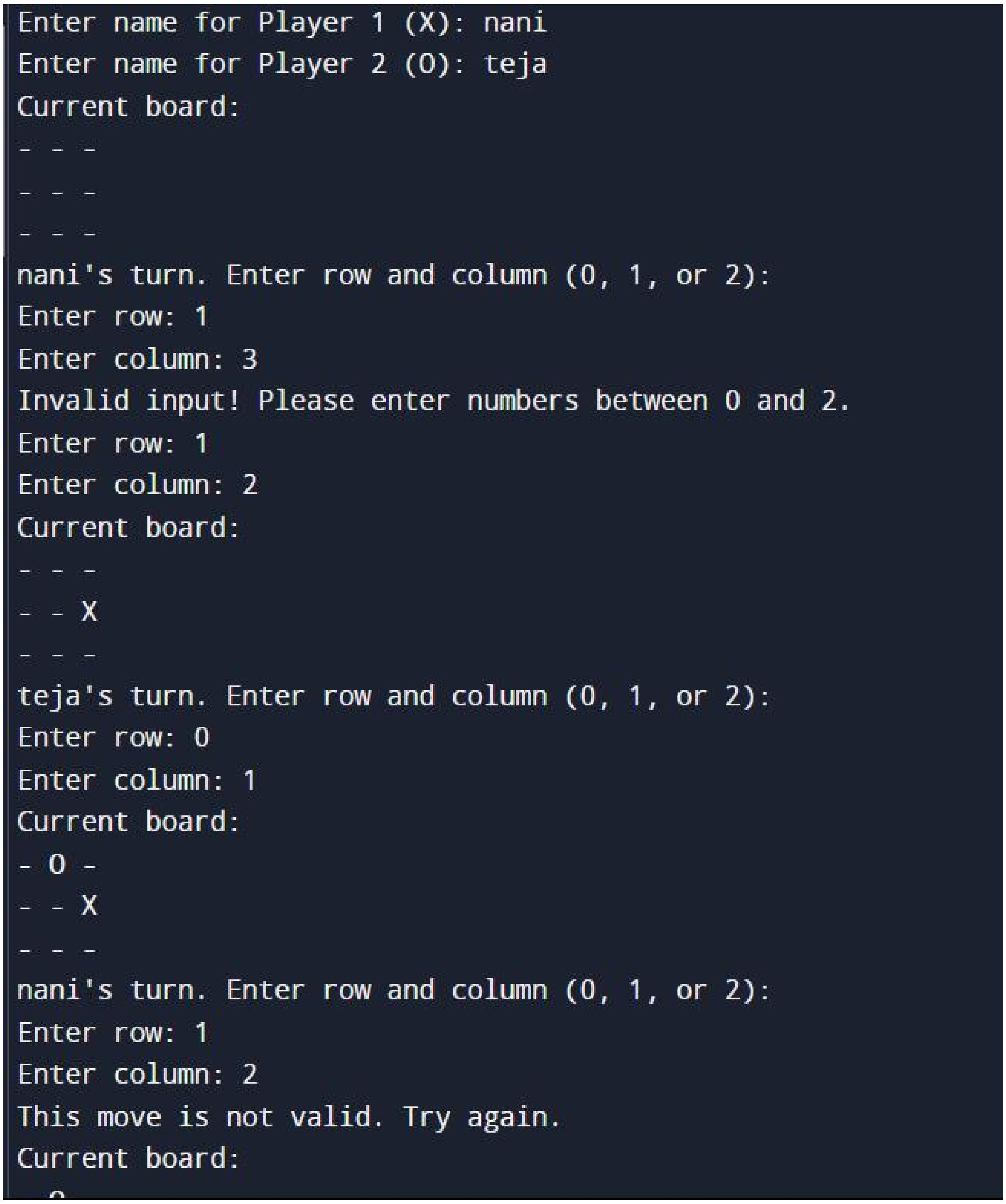
6.

import java.util.Scanner;

```
// Player class
class Player {
    private String name;
    private char symbol;

    public Player(String name, char symbol) {
        this.name = name;
        this.symbol = symbol;
    }

    public String getName() {
        return name;
    }

    public char getSymbol() {
        return symbol;
    }
}

// TicTacToe class
public class TicTacToe {
    private char[][] board;
    private Player player1;
    private Player player2;
    private Player currentPlayer;

    public TicTacToe(Player player1, Player player2) {
        board = new char[3][3];
        this.player1 = player1;
        this.player2 = player2;
        this.currentPlayer = player1;
        initializeBoard();
    }

    private void initializeBoard() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                board[i][j] = '-';
            }
        }
    }

    public void printBoard() {
        System.out.println("Current board:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(board[i][j] + " ");
            }
            System.out.println();
        }
    }

    public boolean makeMove(int row, int col) {
        if (row < 0 || col < 0 || row >= 3 || col >= 3 || board[row][col] != '-') {
            System.out.println("This move is not valid. Try again.");
            return false;
        }
        board[row][col] = currentPlayer.getSymbol();
        return true;
    }
```

```
    public boolean checkWinner() {
        // Check rows and columns
        for (int i = 0; i < 3; i++) {
            if ((board[i][0] == currentPlayer.getSymbol() && board[i][1] == currentPlayer.getSymbol() &&
board[i][2] == currentPlayer.getSymbol()) ||
                (board[0][i] == currentPlayer.getSymbol() && board[1][i] == currentPlayer.getSymbol() &&
board[2][i] == currentPlayer.getSymbol())) {
                return true;
            }
        }
        // Check diagonals
        if ((board[0][0] == currentPlayer.getSymbol() && board[1][1] == currentPlayer.getSymbol() &&
board[2][2] == currentPlayer.getSymbol()) ||
            (board[0][2] == currentPlayer.getSymbol() && board[1][1] == currentPlayer.getSymbol() &&
board[2][0] == currentPlayer.getSymbol())) {
            return true;
        }
        return false;
    }

    public boolean isBoardFull() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (board[i][j] == '-') {
                    return false;
                }
            }
        }
        return true;
    }

    public void switchPlayer() {
        currentPlayer = (currentPlayer == player1) ? player2 : player1;
    }

    public Player getCurrentPlayer() {
        return currentPlayer;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Get player names
        System.out.print("Enter name for Player 1 (X): ");
        Player player1 = new Player(scanner.nextLine(), 'X');

        System.out.print("Enter name for Player 2 (O): ");
        Player player2 = new Player(scanner.nextLine(), 'O');

        // Initialize the game
        TicTacToe game = new TicTacToe(player1, player2);

        // Main game loop
        while (true) {
            game.printBoard();
            System.out.println(game.getCurrentPlayer().getName() + "'s turn. Enter row and column (0, 1, 
or 2): ");

            // Validate input
            int row = -1, col = -1;
            while (true) {
                try {
                    System.out.print("Enter row: ");
                row = scanner.nextInt();
                System.out.print("Enter column: ");
                col = scanner.nextInt();
                if (row >= 0 && row < 3 && col >= 0 && col < 3) break;
                else System.out.println("Invalid input! Please enter numbers between 0 and 2.");
            } catch (Exception e) {
                System.out.println("Invalid input! Please enter numbers between 0 and 2.");
                scanner.nextLine();  // Clear the buffer
            }
        }

        if (game.makeMove(row, col)) {
            if (game.checkWinner()) {
                game.printBoard();
                System.out.println(game.getCurrentPlayer().getName() + " wins!");
                break;
            } else if (game.isBoardFull()) {
                game.printBoard();
                System.out.println("The game is a tie!");
                break;
            } else {
                game.switchPlayer();
            }
        }
    }

    scanner.close();
  }
}
```

Output:



```
Enter name for Player 1 (X): nani
Enter name for Player 2 (O): teja
Current board:
- - -
- - -
- - -
nani's turn. Enter row and column (0, 1, or 2):
Enter row: 1
Enter column: 3
Invalid input! Please enter numbers between 0 and 2.
Enter row: 1
Enter column: 2
Current board:
- - -
- - X
- - -
teja's turn. Enter row and column (0, 1, or 2):
Enter row: 0
Enter column: 1
Current board:
- O -
- - X
- - -
nani's turn. Enter row and column (0, 1, or 2):
Enter row: 1
Enter column: 2
This move is not valid. Try again.
Current board:
```