

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name: MUHAMMAD HAFIZ NABIL BIN ZAINI | | | | | |
| ID Number: AM2307013925 | | | | | |
| Lecturer:  **MDM SITI ROBAYA BINTI JANTAN** | | | | Lab group / Tutorial Group  / Tutor (if applicable)  **SECTION 2** | |
| Course and Course Code:  **PROGRAMMING FOR DATA SCIENCE (SWC2273)** | | | | Submission Date:  **19 / 2 / 2024** | |
| Assignment No. / Title:  **ASSIGNMENT 1** | | | | Extension & Late submission: ALLOWED /  **DISALLOWED** | |
| Assignment type:  **INDIVIDUAL** | | % of Assignment  Mark | | Returning Date: | |
| Penalties:   1. 10 % of the original mark will be deducted for every one-week period after the submission date 2. No work will be accepted after two weeks of the deadline 3. If you were unable to submit the coursework on time due to extenuating circumstances you may be eligible for an extension 4. Extension will not exceed one week | | | | | |
| Declaration: I the undersigned confirm that I have read and agreed to abide by these regulations on plagiarism and cheating. I confirm that this piece of work is my own. I consent to appropriate storage of my work for checking to ensure that there is no plagiarism/ academic cheating.  Signature(s):  Full Name: MUHAMMAD HAFIZ NABIL BIN ZAINI | | | | | |
| This section may be used for feedback or other information: | | | | | |
| **No** | |  | | **Table of Contents** | | **Page** | |
| 1. | | UML Class Diagram | |  | | 1 | |
| 2. | | Coding/ Implementation | |  | | 2 - 5 | |
| 3. | | Complete Output | |  | | 6 - 8 | |

**UML Class Diagram**

|  |
| --- |
| Flower |
| - name : String  - color : String  - price : double  - quantity : int |
| + Flower():  + Flower(String name,String color, double price, int quantity):  + setFlower(name : String, color : String, price : double, Quantity :int) : void  + setName(name : String) :void  + setColor(color : String) : void  + setPrice(price : double) :void  + setQuantity(quantity :int) : void  + getName() : String  + getColor() : String  + getPrice() : double  + getQuantity() : int  + toString() : String |

**Coding / Implementation**

**Flower**

/\*\*

\* program description: LabWork Flower

\*

\* programmer: MUHAMMAD HAFIZ NABIL BIN ZAINI

\* Date: 19 February 2024

\*/

import java.util.Scanner;

public class Flower

{

// instance variables - replace the example below with your own

private String name;

private String color;

private double price;

private int quantity;

// Normal constructor with parameter

public Flower(String name, String color, double price, int quantity ){

this.name = name;

this.color = color;

this.price = price;

this.quantity = quantity;

}

//Constructor without parameter

public Flower(){

this.name = "";

this.color = "";

this.price = 10;

this.quantity = 20;

}

//retrievers (accessors / getters) for each data member

public String getName(){

return name;

}

public String getColor(){

return color;

}

public double getPrice(){

return price;

}

public int getQuantity(){

return quantity;

}

//Mutators (setters) that sets all data members

public void setName(String name) {

this.name = name;

}

public void setColor(String color) {

this.color = color;

}

public void setPrice(double price) {

this.price= price;

}

public void setQuantity(int quantity) {

this.quantity= quantity;

}

@Override

public String toString() {

return "Flower Details:\n" +

"Name ='" + name + "\n" +

"Color =" + color + "\n" +

"Price =" + price + "\n" +

"Quantity =" + quantity;

}

//display flower name, color, price and quantity

public static void main(String[] args) {

Flower lavender = new Flower("Lavender", "Purple", 15.00, 30);

System.out.println(lavender);//Display flower details using toString()

}

}

**FlowerShop**

/\*\*

\* program description: LabWork FlowerShop

\*

\* programmer: MUHAMMAD HAFIZ NABIL BIN ZAINI

\* Date: 19 February 2024

\*/

import java.util.Scanner;

public class FlowerShop {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Flower[] flowerInventory = new Flower[10];

// Prompt user to enter data for each flower

for (int i = 0; i < flowerInventory.length; i++) {

System.out.println("Enter details for Flower #" + (i + 1));

System.out.print("Name: ");

String name = scanner.nextLine();

System.out.print("Color: ");

String color = scanner.nextLine();

System.out.print("Price: $");

double price = scanner.nextDouble();

System.out.print("Quantity: ");

int quantity = scanner.nextInt();

scanner.nextLine(); // Consume newline

flowerInventory[i] = new Flower(name, color, price, quantity);

}

// Display all flower details

System.out.println("\nFlower Inventory:");

for (Flower flower : flowerInventory) {

System.out.println(flower);

}

// Calculate and display total value of the flower inventory

double totalValue = 0.0;

for (Flower flower : flowerInventory) {

totalValue += flower.getPrice() \* flower.getQuantity();

}

System.out.println("\nTotal Inventory Value: $" + totalValue);

// Search for a flower by name

System.out.print("\nEnter a flower name to search: ");

String searchName = scanner.nextLine();

for (Flower flower : flowerInventory) {

if (flower.getName().equalsIgnoreCase(searchName)) {

System.out.println("Found flower:");

System.out.println(flower);

break;

}

}

// Restock a specific flower

System.out.print("\nEnter the index of the flower to restock (0-9): ");

int index = scanner.nextInt();

System.out.print("Enter the additional quantity: ");

int additionalQuantity = scanner.nextInt();

flowerInventory[index].setQuantity(flowerInventory[index].getQuantity() + additionalQuantity);

// Display updated flower details

System.out.println("\nUpdated Flower Inventory:");

for (Flower flower : flowerInventory) {

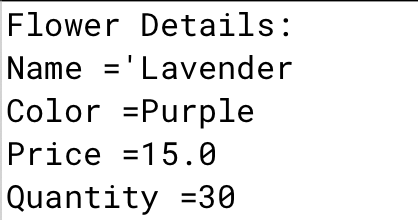
System.out.println(flower);

}

}

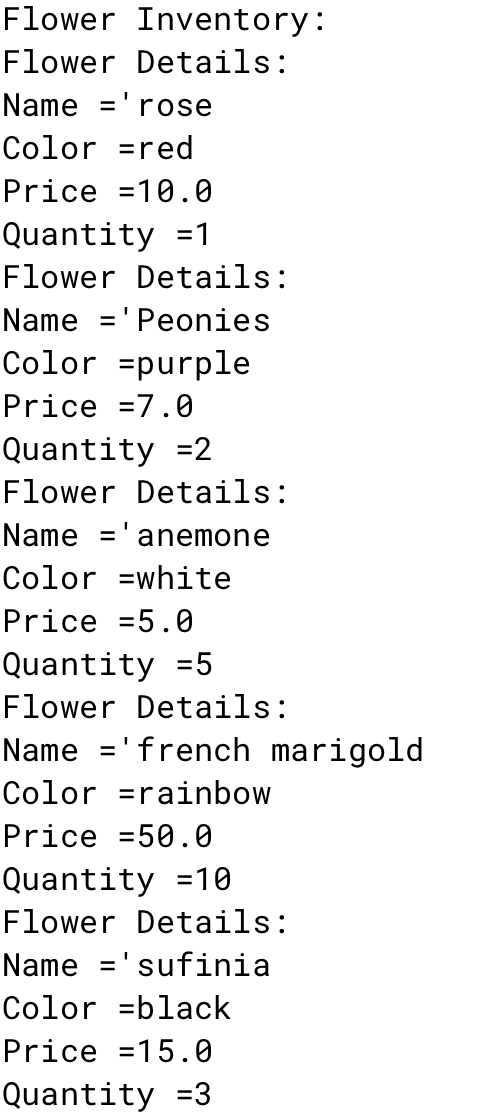
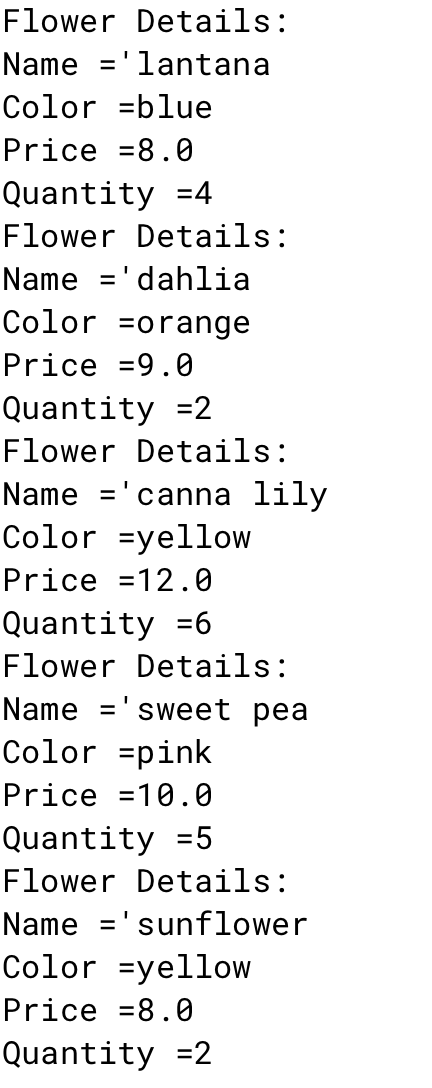
}

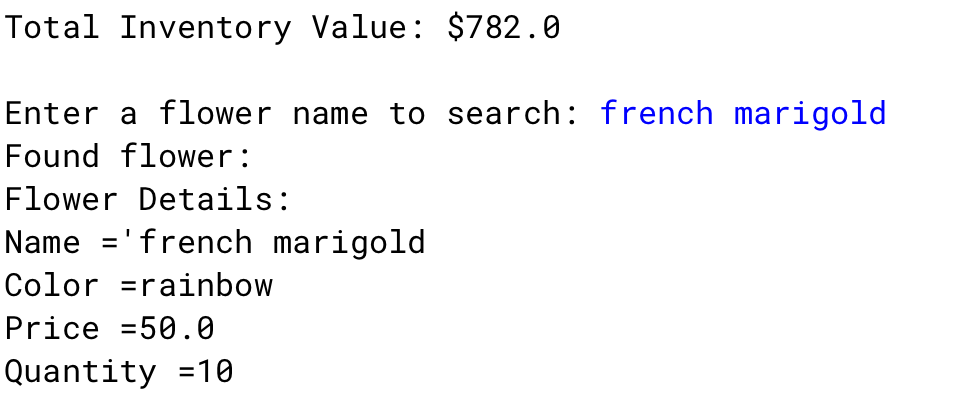
**Complete Output**

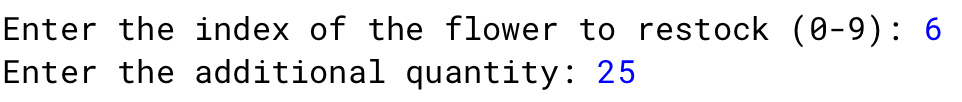
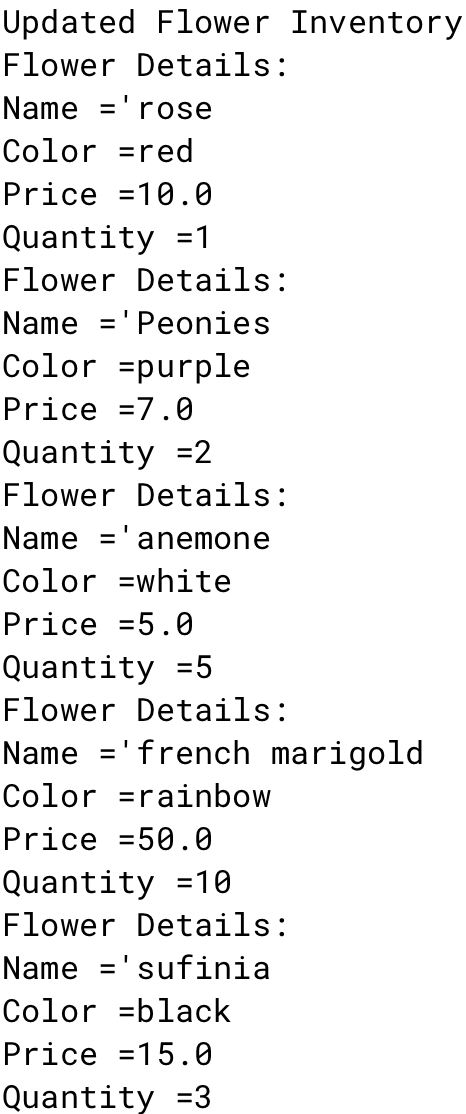
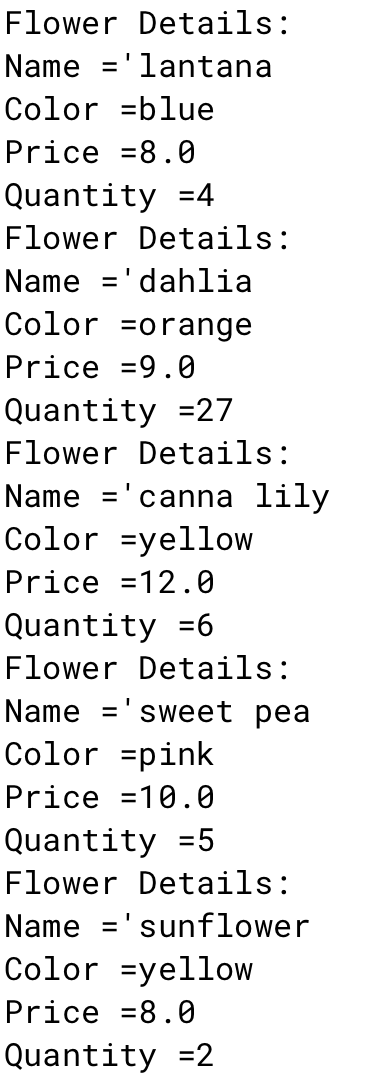
**FLOWER**

**FLOWERSHOP**

****

****

****

****