Billing System Java

By:- Assignwise

Contents

1. Introduction

The Java program for the local handicraft store in Patan is designed to provide an efficient and user-friendly billing system for customers. The program adopts an object-oriented approach, utilizing classes and objects to handle customer orders and generate bills. It starts by displaying a menu of available items for sale, enabling customers to make orders based on their preferences. The system can manage orders and billing for up to 5 customers simultaneously, ensuring smooth operation during busy times.

When a customer places an order, the program calculates the total bill, which comprises three components: the amount going to the local charity, the amount allocated to the store, and the total payment the customer needs to make. The program ensures transparency by displaying the charity contribution, which is set at 12% of the sold amount, aiding the local community and fostering a sense of social responsibility. The billing process is streamlined, allowing customers to quickly see the breakdown of their purchase and encouraging them to contribute to a noble cause.

Overall, this object-oriented billing system offers a seamless shopping experience for customers at the handicraft store in Patan. It efficiently manages multiple orders and generates transparent bills, reflecting the charity contribution and the store's earnings. By incorporating these features, the program not only enhances the store's operations but also fosters a sense of social responsibility among customers by supporting the local charity. This simple yet effective billing system plays a vital role in promoting the store's reputation and ensuring satisfied and socially conscious customers.

2. Detailed Algorithm

2.1 Product class

Step 1: Start

Step 2: Declare a class named "Product".

Step 3: Declare private instance variables within the class:
i. Declare a private String variable `productName`.
ii. Declare a private int variable `productPrice`.

Step 4: Define a constructor for the class "Product":
i. Receive two parameters: a String `name` and an int `price`.
ii. Set the `productName` instance variable to the value of the `name` parameter.
iii. Set the `productPrice` instance variable to the value of the `price` parameter.

Step 5: Define a public method `getName()`:
i. Return the value of the `productName` instance variable.

Step 6: Define a public method `getPrice()`:
i. Return the value of the `productPrice` instance variable.

Step 7: End

2.2 Customer class

Step 1: Start

Step 2: Create a Java class named `Customer`.

Step 3: Declare private instance variables:
- `customerName` of type String to store the customer's name.
- `customerCount` of type int (static) to maintain a count of customers.
- `customerNumber` of type int to store the customer's unique ID.

Step 4: Create a constructor `Customer(String name)`:
- Initialize `customerName` with the provided name.
- Initialize `customerNumber` with the current value of `customerCount` and then increment `customerCount`.

Step 5: Create a method `getName()`:
- Return the value of `customerName`.

Step 6: Create a method `getCustomerNumber()`:
- Return the value of `customerNumber`.

Step 7: End

2.3 HandicraftStore class

Step 1: Start

Step 2: Create a Java class named HandicraftStore.

Step 3: Create a private class named 'Product' with private fields 'name' and 'price'. Include a constructor to initialize these fields.

Step 4: Inside the 'Product' class, create public methods 'getName()' and 'getPrice()' to retrieve the name and price of a product.

Step 5: Initialize the 'products' array with instances of the 'Product' class, each containing a product name and its corresponding price:

- Initialize 'products' array with length 10
- Create 10 instances of 'Product' with the following names and prices:
  -"Handmade Bag " – 3500
  - "Glay Art" – 4000
  -“ Happy Buddha" – 3000
  -“ Batik Painting" – 5500
  -“Beaded Jewellery" – 7500
  -“Glass Painting" – 2500
  -“Pachisi" – 2000
  -“Floral Candle" – 1500
  -“Painted Wooden Art" – 4800
  -"Bamboo Painting" - 4500

Step 6: Create a private final double variable 'CHARITY_PERCENTAGE' and initialize it with 0.12.

Step 7: Define the constructor 'HandicraftStore()' where you initialize the 'products' array with the products created in Step 4.

Step 8: Define a public method 'takeCustomerOrders()' that takes customer orders as input and returns an array 'selectedProductsCount' representing the count of selected products.

a. Create a Scanner object 'input' to read user input.

b. Initialize an 'int' array 'selectedProductsCount' with length equal to the number of products.

c. Display the menu by iterating through the 'products' array and showing the product name and price.

d. Start a loop:

- Prompt the user to enter the product number (1 to finish).
- Read the user's selected option.
- If the selected option is 0, exit the loop.
- If the selected option is valid (between 1 and number of products), prompt the user to enter the number of pieces for the selected product.

- Update the 'selectedProductsCount' array with the selected number of pieces for the product.
- If the selected option is invalid, show an error message.

Step 9: Define a public method 'calculateTotalBill(int[] selectedProductsCount)' that calculates the total bill and returns an array 'double[]' containing total price, charity amount, and store amount.

a. Initialize a variable 'totalPrice' to 0.

b. Iterate through the 'products' array:

- Add the product's price multiplied by the count from 'selectedProductsCount' to 'totalPrice'.

c. Calculate the 'charityAmount' by multiplying 'totalPrice' with 'CHARITY_PERCENTAGE'.

d. Calculate 'storeAmount' by subtracting 'charityAmount' from 'totalPrice'.

e. Return an array containing 'totalPrice', 'charityAmount', and 'storeAmount'.

Step 10: Define a public method 'getProducts()' that returns the 'products' array.

Step 11: End

2.4 Bill class

Step 1: Start

Step 2: Create a class named "Bill"

Step 3: Define private instance variables within the "Bill" class:
- customer: of type Customer
- totalPrice: of type double
- charityAmount: of type double
- storeAmount: of type double
- orderedProducts: an array of type Product
- orderedProductCounts: an array of type int
- orderedProductPrices: an array of type double

Step 4: Define a constructor for the "Bill" class that accepts parameters:
- customer: Customer
- totalPrice: double
- charityAmount: double
- storeAmount: double
- orderedProducts: Product[]
- orderedProductCounts: int[]
- orderedProductPrices: double[]

Initialize the instance variables using the provided parameters.

Step 5: Define a method named "displayCustomerBill" within the "Bill" class:

i. Print the store header and welcome messages.

ii. Print the customer's information and bill details:
- Customer number
- Customer name

iii. Print the details of ordered products using a loop:

For each product in orderedProducts:
- Print product name
- Print ordered quantity
- Print product price

iv. Print the total price, charity amount, and store amount.

Step 6: Create an instance of the "Bill" class by passing appropriate values to the constructor.

Step 7: Call the "displayCustomerBill" method on the created "Bill" instance to display the bill details.

Step 8: Stop

2.5 Main class

Step 1: Start

Step 2: Create a Java class named Main.

a. Define the 'main' method.

Step 3: Instantiate necessary objects and variables.

a. Create a HandicraftStore object named 'store'.

b. Instantiate a Scanner named 'input' to read user input from the console.

c. Define an integer variable 'numberOfCustomers' and set it to the desired number of customers (e.g., 5).

Step 4: Iterate through each customer.

a. Loop 'i' from 0 to 'numberOfCustomers - 1':

i. Display "Enter customer name: ".

ii. Read and store customer's name in 'customerName'.

iii.Create a Customer object named 'customer' with 'customerName'.

iv.Get the selected product counts from the store using 'store.takeCustomerOrders()'.

v.Calculate the total billamounts using 'store.calculateTotalBill (selectedProductsCount)'.

vi. Initialize variables for tracking ordered products:

- Create arrays 'orderedProducts', 'orderedProductCounts', and 'orderedProductPrices' of suitable length.
- Initialize 'totalOrderedProducts' as 0.

vii. Loop 'j' through the range of product array:

- If 'selectedProductsCount[j]' is greater than 0:
- Add the product to 'orderedProducts' array.
- Add the count to 'orderedProductCounts' array.
- Calculate and add the price to 'orderedProductPrices' array.
- Increment 'totalOrderedProducts'.

viii.Create arrays 'finalOrderedProducts', 'finalOrderedProductCounts', and 'finalOrderedProductPrices' using 'Arrays.copyOf()' to trim unused space in the arrays.

ix. Create a Bill object named 'bill' with customer, bill amounts, and ordered product information.

x. Call 'bill.displayCustomerBill()' to display the customer's bill.

xi. Display "Thank you for choosing Himalayan Handicraft Store." and "Visit us again soon!".

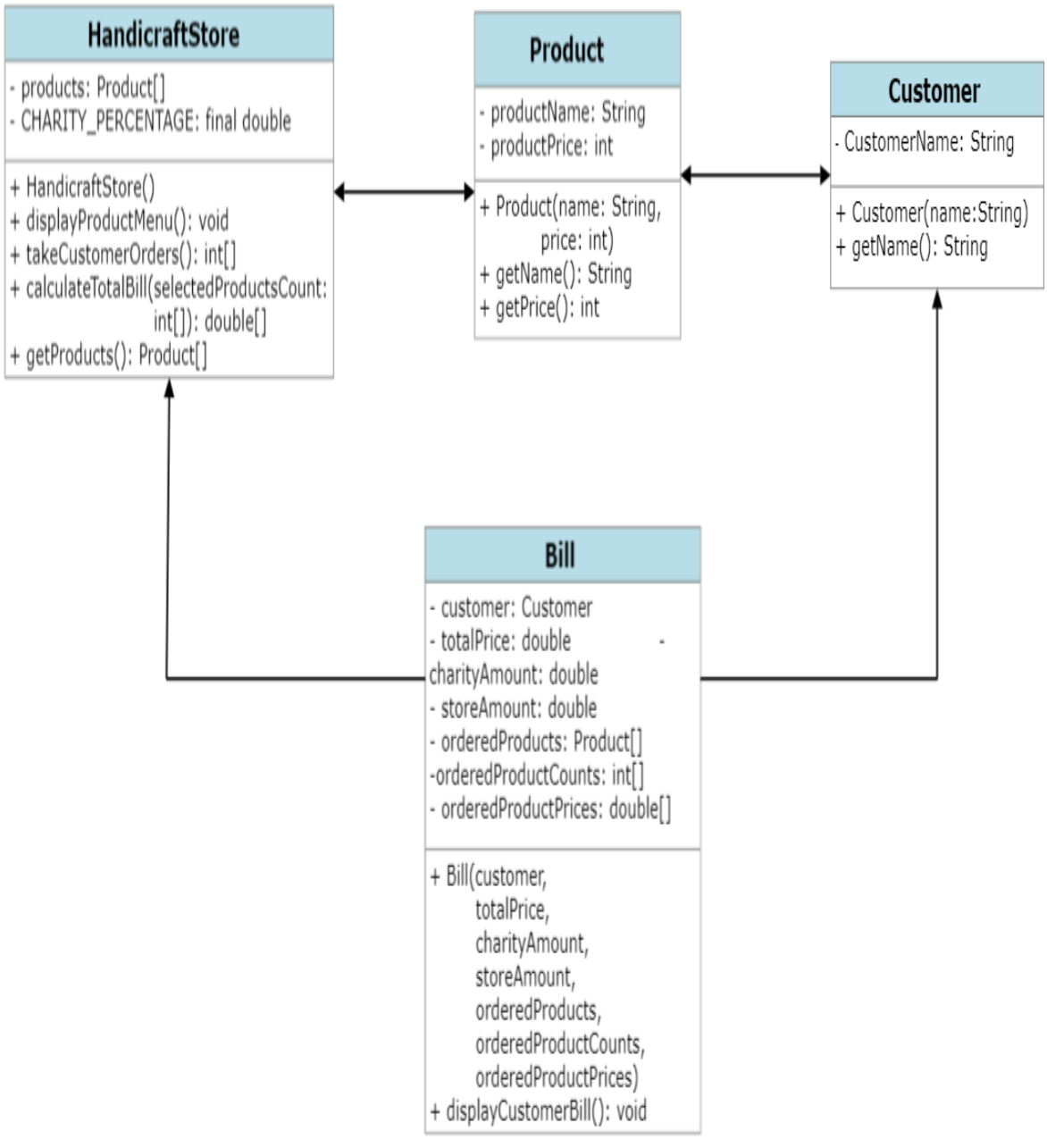
xii. Display "Do you want to continue with the next customer? (yes/no): ".

xiii. Read and store the user's choice in 'repeat'.

xiv. If 'repeat' is not equal to "yes", break the loop.

Step 5: End

3. UML Classes



4. Source Code

4.1 Product.java

Product.java > ...

```java
import java.util.Scanner;

class Product {
    private String productName;
    private int productPrice;

    public Product(String name, int price) {
        this.productName = name;
        this.productPrice = price;
    }

    public String getName() {
        return productName;
    }

    public int getPrice() {
        return productPrice;
    }
}
```

4.2 Customer.java

Customer.java > ...

```java
import java.util.Scanner;

class Customer {
    private String customerName;
    private static int customerCount = 1;
    private int customerNumber;

    public Customer(String name) {
        this.customerName = name;
        this.customerNumber = customerCount++;
    }

    public String getName() {
        return customerName;
    }

    public int getCustomerNumber() {
        return customerNumber;
    }
}
```

4.3 HandicraftStore.java

HandicraftStore.java > ...

```java
import java.util.Scanner;

class HandicraftStore {
    private Product[] products;
    private final double CHARITY_PERCENTAGE = 0.12;

    public HandicraftStore() {
        products = new Product[10];
        products[0] = new Product(name:"Handmade Bag ", price:350);
        products[1] = new Product(name:"Glay Art", price:400);
        products[2] = new Product(name:"Happy Buddha", price:300);
        products[3] = new Product(name:"Batik Painting", price:550);
        products[4] = new Product(name:"Beaded Jewellery", price:750);
        products[5] = new Product(name:"Glass Painting", price:250);
        products[6] = new Product(name:"Pachisi", price:200);
        products[7] = new Product(name:"Floral Candle", price:150);
        products[8] = new Product(name:"Painted Wooden Art", price:480);
        products[9] = new Product(name:"Bamboo Painting", price:450);
    }

    public int[] takeCustomerOrders() {
        Scanner input = new Scanner(System.in);
        int[] selectedProductsCount = new int[products.length];

        System.out.println(x:"Menu:");
        for (int i = 0; i < products.length; i++) {
            System.out.println((i + 1) + ". " + products[i].getName() + " - Rs " + products[i].getPrice());
        }

        while (true) {
            System.out.print(s:"Enter the product number (1 to finish): ");
            int selectedOption = input.nextInt();

            if (selectedOption == 0) {
                break;
            }

            if (selectedOption > 0 && selectedOption <= products.length) {
                System.out.print("Enter the number of pieces for " + products[selectedOption - 1].getName() + ": ");
                int pieces = input.nextInt();
                selectedProductsCount[selectedOption - 1] += pieces;
            } else {
                System.out.println(x:"Invalid product number. Please try again.");
            }
        }

        return selectedProductsCount;
    }

    public double[] calculateTotalBill(int[] selectedProductsCount) {
        int totalPrice = 0;
        for (int i = 0; i < products.length; i++) {
            totalPrice += products[i].getPrice() * selectedProductsCount[i];
        }

        double charityAmount = totalPrice * CHARITY_PERCENTAGE;
        double storeAmount = totalPrice - charityAmount;

        return new double[] { totalPrice, charityAmount, storeAmount };
    }

    public Product[] getProducts() {
        return products;
    }
}
```

4.4 Bill.java

Bill.java > ...

```java
import java.util.Scanner;

class Bill {
    private Customer customer;
    private double totalPrice;
    private double charityAmount;
    private double storeAmount;
    private Product[] orderedProducts;
    private int[] orderedProductCounts;
    private double[] orderedProductPrices;

    public Bill(Customer customer, double totalPrice, double charityAmount, double storeAmount,
            Product[] orderedProducts, int[] orderedProductCounts, double[] orderedProductPrices) {
        this.customer = customer;
        this.totalPrice = totalPrice;
        this.charityAmount = charityAmount;
        this.storeAmount = storeAmount;
        this.orderedProducts = orderedProducts;
        this.orderedProductCounts = orderedProductCounts;
        this.orderedProductPrices = orderedProductPrices;
    }

    public void displayCustomerBill() {

        System.out.println("\n#####################################");
        System.out.println("      Himalayan Handicrafts Store     ");
        System.out.println("#####################################");
        System.out.println("           Namaste        ");
        System.out.println("Welcome to Himalayan Handicraft Store.");
        System.out.println("\n=====================================");
        System.out.println("         Customer Bill #" + customer.getCustomerNumber());
        System.out.println("         Customer Name: " + customer.getName());
        System.out.println("=====================================");
        System.out.println("\nOrdered Products:");
        for (int i = 0; i < orderedProducts.length; i++) {
            System.out.println(orderedProducts[i].getName() + " - " + orderedProductCounts[i] + " pieces - Rs "
                    + orderedProductPrices[i]);
        }
        System.out.println("\nTotal Price: Rs " + totalPrice);
        System.out.println("Charity Amount: Rs " + charityAmount);
        System.out.println("Store Amount: Rs " + storeAmount);
    }
}
```

4.5 Main.java

Main.java

E: > Pappu > Main.java

```java
import java.util.Arrays;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        HandicraftStore store = new HandicraftStore();
        Scanner input = new Scanner(System.in);

        int numberOfCustomers = 5;
        for (int i = 0; i < numberOfCustomers; i++) {
            System.out.print("\nEnter customer name: ");
            String customerName = input.nextLine();
            Customer customer = new Customer(customerName);

            int[] selectedProductsCount = store.takeCustomerOrders();

            double[] amounts = store.calculateTotalBill(selectedProductsCount);

            int totalOrderedProducts = 0;
            Product[] orderedProducts = new Product[store.getProducts().length];
            int[] orderedProductCounts = new int[store.getProducts().length];
            double[] orderedProductPrices = new double[store.getProducts().length];

            for (int j = 0; j < store.getProducts().length; j++) {
                if (selectedProductsCount[j] > 0) {
                    orderedProducts[totalOrderedProducts] = store.getProducts()[j];
                    orderedProductCounts[totalOrderedProducts] = selectedProductsCount[j];
                    orderedProductPrices[totalOrderedProducts] = selectedProductsCount[j]
                            * store.getProducts()[j].getPrice();
                    totalOrderedProducts++;
                }
            }
            Product[] finalOrderedProducts = Arrays.copyOf(orderedProducts, totalOrderedProducts);
            int[] finalOrderedProductCounts = Arrays.copyOf(orderedProductCounts, totalOrderedProducts);
            double[] finalOrderedProductPrices = Arrays.copyOf(orderedProductPrices, totalOrderedProducts);

            Bill bill = new Bill(customer, amounts[0], amounts[1], amounts[2], finalOrderedProducts,
                    finalOrderedProductCounts, finalOrderedProductPrices);

            bill.displayCustomerBill();

            System.out.println(x:"\nThank you for choosing Himalayan Handicraft Store.");
            System.out.println(x:"Visit us again soon!");

            System.out.print(s:"\nDo you want to continue with the next customer? (yes/no): ");
            String repeat = input.nextLine();
            if (!repeat.equalsIgnoreCase(anotherString:"yes")) {
                break;
            }
        }
    }
}
```

5. Test plan and test data to be used with testing

Customer Number	Customer Name	Product Number	Pieces	Total Price	Charity Amount	Store Amount
1	Pappu Yadav	6	1	Rs 7000.0	Rs 840.0	Rs 6160.0
		10	1			
2	Sudha Magar	1	1	Rs 18500.0	Rs2220.0	Rs16280.0
		5	2			
1	Neha Sharma	2	1	Rs 13300.0	Rs 1596.0	Rs11704.0
		8	3			
		9	1			
2	Kusum Thapa	3	2	Rs 8000.0	Rs 960.0	Rs 7040.0
		7	1			
3	Reeya Jw	5	1	Rs 7500.0	Rs 900.0	Rs 6600.0
4	Sabina Rana	4	1	Rs 10000.0	Rs 1200.0	Rs 8800.0
		10	1			
5	Sesema Limbu	2	2	Rs 17600.0	Rs 2112.0	Rs 15488.0
		9	2			
1	Chandani Shrestha	1	1	Rs 8500.0	Rs 1020.0	Rs 7480.0
		3	1			
		7	1			
1	Rojina Chapagain	4	1	Rs 16500.0	Rs 1980.0	Rs 14520.0
		6	1			
		10	1			
		2	1			
2	Kareena Khadka	3	3	Rs 13500.0	Rs 1620.0	Rs 11880.0
		8	3			

6. Output Screenshot

```
Enter customer name: Pappu Yadav
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 6
Enter the number of pieces for Glass Painting: 1
Enter the product number (1 to finish): 10
Enter the number of pieces for Bamboo Painting: 1
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #1
         Customer Name: Pappu Yadav
========================================

Ordered Products:
Glass Painting - 1 pieces - Rs 2500.0
Bamboo Painting - 1 pieces - Rs 4500.0

Total Price: Rs 7000.0
Charity Amount: Rs 840.0
Store Amount: Rs 6160.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): yes
```

```
Enter customer name: Sudha Magar
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 1
Enter the number of pieces for Handmade Bag : 1
Enter the product number (1 to finish): 5
Enter the number of pieces for Beaded Jewellery: 2
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #2
         Customer Name: Sudha Magar
========================================

Ordered Products:
Handmade Bag  - 1 pieces - Rs 3500.0
Beaded Jewellery - 2 pieces - Rs 15000.0

Total Price: Rs 18500.0
Charity Amount: Rs 2220.0
Store Amount: Rs 16280.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): no
```

```
Enter customer name: Neha Sharma
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 2
Enter the number of pieces for Glay Art: 1
Enter the product number (1 to finish): 8
Enter the number of pieces for Floral Candle: 3
Enter the product number (1 to finish): 9
Enter the number of pieces for Painted Wooden Art: 1
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #1
         Customer Name: Neha Sharma
========================================

Ordered Products:
Glay Art - 1 pieces - Rs 4000.0
Floral Candle - 3 pieces - Rs 4500.0
Painted Wooden Art - 1 pieces - Rs 4800.0

Total Price: Rs 13300.0
Charity Amount: Rs 1596.0
Store Amount: Rs 11704.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): yes
```

```
Enter customer name: Kusum Thapa
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 3
Enter the number of pieces for Happy Buddha: 2
Enter the product number (1 to finish): 7
Enter the number of pieces for Pachisi: 1
Enter the product number (1 to finish): 0

##########################################
      Himalayan Handicrafts Store
##########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #2
         Customer Name: Kusum Thapa
========================================

Ordered Products:
Happy Buddha - 2 pieces - Rs 6000.0
Pachisi - 1 pieces - Rs 2000.0

Total Price: Rs 8000.0
Charity Amount: Rs 960.0
Store Amount: Rs 7040.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!
```

```
Enter customer name: Reeya Jw
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 5
Enter the number of pieces for Beaded Jewellery: 1
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #3
         Customer Name: Reeya Jw
========================================

Ordered Products:
Beaded Jewellery - 1 pieces - Rs 7500.0

Total Price: Rs 7500.0
Charity Amount: Rs 900.0
Store Amount: Rs 6600.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): yes
```

```
Enter customer name: Sabina Rana
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 4
Enter the number of pieces for Batik Painting: 1
Enter the product number (1 to finish): 10
Enter the number of pieces for Bamboo Painting: 1
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #4
         Customer Name: Sabina Rana
========================================

Ordered Products:
Batik Painting - 1 pieces - Rs 5500.0
Bamboo Painting - 1 pieces - Rs 4500.0

Total Price: Rs 10000.0
Charity Amount: Rs 1200.0
Store Amount: Rs 8800.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): yes
```

```
Enter customer name: Sesema Limbu
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 2
Enter the number of pieces for Glay Art: 2
Enter the product number (1 to finish): 9
Enter the number of pieces for Painted Wooden Art: 2
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
           Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #5
         Customer Name: Sesema Limbu
========================================

Ordered Products:
Glay Art - 2 pieces - Rs 8000.0
Painted Wooden Art - 2 pieces - Rs 9600.0

Total Price: Rs 17600.0
Charity Amount: Rs 2112.0
Store Amount: Rs 15488.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): yes
```

```
Enter customer name: Chandani Shrestha
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 1
Enter the number of pieces for Handmade Bag : 1
Enter the product number (1 to finish): 3
Enter the number of pieces for Happy Buddha: 1
Enter the product number (1 to finish): 7
Enter the number of pieces for Pachisi: 1
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #1
         Customer Name: Chandani Shrestha
========================================

Ordered Products:
Handmade Bag  - 1 pieces - Rs 3500.0
Happy Buddha - 1 pieces - Rs 3000.0
Pachisi - 1 pieces - Rs 2000.0

Total Price: Rs 8500.0
Charity Amount: Rs 1020.0
Store Amount: Rs 7480.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): no
```

```
Enter customer name: Rojina Chapagain
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 4
Enter the number of pieces for Batik Painting: 1
Enter the product number (1 to finish): 6
Enter the number of pieces for Glass Painting: 1
Enter the product number (1 to finish): 10
Enter the number of pieces for Bamboo Painting: 1
Enter the product number (1 to finish): 2
Enter the number of pieces for Glay Art: 1
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #1
         Customer Name: Rojina Chapagain
========================================

Ordered Products:
Glay Art - 1 pieces - Rs 4000.0
Batik Painting - 1 pieces - Rs 5500.0
Glass Painting - 1 pieces - Rs 2500.0
Bamboo Painting - 1 pieces - Rs 4500.0

Total Price: Rs 16500.0
Charity Amount: Rs 1980.0
Store Amount: Rs 14520.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): yes
```

```
Enter customer name: Kareena Khadka
Menu:
1. Handmade Bag  - Rs 3500
2. Glay Art - Rs 4000
3. Happy Buddha - Rs 3000
4. Batik Painting - Rs 5500
5. Beaded Jewellery - Rs 7500
6. Glass Painting - Rs 2500
7. Pachisi - Rs 2000
8. Floral Candle - Rs 1500
9. Painted Wooden Art - Rs 4800
10. Bamboo Painting - Rs 4500
Enter the product number (1 to finish): 3
Enter the number of pieces for Happy Buddha: 3
Enter the product number (1 to finish): 8
Enter the number of pieces for Floral Candle: 3
Enter the product number (1 to finish): 0

########################################
      Himalayan Handicrafts Store
########################################
            Namaste
Welcome to Himalayan Handicraft Store.

========================================
         Customer Bill #2
         Customer Name: Kareena Khadka
========================================

Ordered Products:
Happy Buddha - 3 pieces - Rs 9000.0
Floral Candle - 3 pieces - Rs 4500.0

Total Price: Rs 13500.0
Charity Amount: Rs 1620.0
Store Amount: Rs 11880.0

Thank you for choosing Himalayan Handicraft Store.
Visit us again soon!

Do you want to continue with the next customer? (yes/no): no
```

7. Conclusion

It was a good learning opportunity for me to create a billing system for a neighborhood handicraft shop in Patan. I developed a thorough understanding of object-oriented programming and effectively created functionalities for order handling and product administration. Stakeholders were impressed by the system's adaptability in managing financial transactions, including gifts to charities. To build a productive program, I used a variety of programming techniques like loops, switch-case statements, and constants. I learned the value of meticulous attention to detail and problem-solving abilities via testing and debugging. My knowledge in computer science and software development has significantly increased as a result of this project.

References

1. University (Slides)
   (August 15, 2023)

2. Java coding
   Topper (Online)
   https://topper.com/gate/introduction-to-c-programming/
   (August 24, 2023)

3. UML Classes diagram
   Smart Draw(Online)
   https://smartdraw/umlclassdiagram/
   (August 25 , 2023)

4. Online java compiler
   https://www.jdoodle.com/online-java-compiler/
   (August 25, 2023)

Marking Scheme

20% Design

- UML class diagram with essential attributes and operations.
- Pseudocode/algorithm to illustrate your solution logic

15% quality of Java program implementation attributes and operations

- Data structure and variable allocation.
- Clear and straightforward code to solve the problem.

20% Test plan and cases

- Test cases and test data
- Output screenshots

40% Correctness

- Code produces correct result

10% Documentation

- Well-structured with content page, introduction design, test plan, program listing, screen shots and conclusion with reference.