A Project Report

BSCY Discount Calculator for E-Commerce 2025

on

Discount Calculator for E-Commerce



by

Shayan Ullah Arif BCY243014

Laiba Waheed Abbasi BCY243008

A Project Report submitted to the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

in partial fulfillment of the requirements for the degree of

BACHELORS OF SCIENCE IN ELECTRICAL ENGINEERING

Faculty of Engineering

Capital University of Science & Technology,

Islamabad

Content Page

1. Introduction Page 1
2. Problem Definition Page 2
3. Methodologies Used Page 3
4. Code Page 4
5. Results Page 6
6. Conclusion Page 7
7. References Page 8

## Introduction

In today's competitive retail and e-commerce environment, calculating the final price after apptying discounts is essential for maintaining transparent pricing and customer satisfaction. However, performing these calculations manually can often lead to errors, especially when dealing with multiple pricing strategies. This project aims to automate the process of discount calculation, ensuring accuracy, efficiency, and ease of use.

The goat of our project is to develop a simple yet effective C++ program that allows users to calculate the final price of a product based on two types of discounts: percentage-based and fixed-amount discounts. By implementing this system, businesses can speed up their discount application processes, reducing human error and improving customer experience.

In this report, we wilt explain the problem that ted to the creation of the Discount Calculation System, the methodologies used to develop the system, provide the complete source code, present the results from various test cases, and conclude with some future directions for improvement.

1. **Problem** Definition

The problem arises when businesses need to apply discounts to products, either as a percentage of the original price or as a fixed amount. Depending on the discount type, calculating the final price

after applying the discount can involve different steps, leading to potential human errors.

For instance:

* + **Percentage Discount: A** discount given as a percentage of the product's price (e.g., 20% off).
  + **Fixed Discount: A** discount given as a fixed monetary amount (e.g., $15 off).

Without an efficient tool, employees might make mistakes during these calculations, resulting in incorrect pricing and potentially affecting business profits or customer trust. This issue becomes more pronounced in larger businesses where multiple discounts are applied across numerous products.

This project seeks to solve this problem by automating the calculation process, allowing users to enter a product price and select the type of discount they wish to apply. The system will then compute and display the final price accurately.

1. Methodologies Used

The development of this system followed a structured approach, leveraging standard programming techniques and best practices:

1. **Problem Analysis & Requirement Gathering:**

The first step was to understand the core problem: how to automate the calculation of discounts. We identified the need for two primary discount methods: percentage-based and fixed-amount discounts. This laid the foundation for designing the program.

1. **System Design**:

The program was designed to be simple and user-friendly. The system asks for the original price of the product, then prompts the user to choose between two discount options. After receiving the user's input, the program computes the discount based on the selected type and displays the final price.

1. **Implementation Using C++:**

The code was written using C++ due to its efficiency and familiarity in handling user inputs and arithmetic operations. The following constructs were used:

* 1. Variables to store the product price, discount

percentage, and fixed discount value.

* 1. Conditionals (if-e\se) to distinguish between the two discount types and calculate the final price accordingly.
  2. Input and Output for seamless interaction with the user, providing prompts and displaying results.

1. **Testing and Debugging**:

Multiple test cases were executed to ensure the program works correctly under various conditions:

* 1. Normal inputs with valid percentages and fixed amounts.
  2. Handling of invalid discount types to ensure the program doesn't crash.
  3. Verifying accuracy of the final price calculations with edge cases (e.g., 0% discount or a very large fixed discount).

1. Code

Here is the complete code for the Discount Calculation System:

#include <iostream> using namespace std;

int main()(

float price, percentageDiscount, fixedDiscount;

int discountType;

cout << "Enter the price of the product: "; cin >> price;

cout << "Choose the type of discount: " << endl;

cout << "1. Percentage Discount" << endl; cout << "2. Fixed Amount Discount" << endI; cin >> discountType;

float finalPrice = 0; *II* Initialize finalPrice to a default value

if (discountType == 1)(

cout << "Enter the discount percentage: "; cin >> percentageDiscount;

finalPrice = price - (price \* (percentageDiscount / 100));

) else if (discountType == 2)(

cout << "Enter the fixed discount amount: "; cin >> fixedDiscount;

finalPrice = price - fixedDiscount;

) else (

cout << "Invalid discount type chosen!" << endl;

*II* Only print finalPrice if it's properly calculated if (discountType == 1 || discountType == 2) (

cout << ”The final price after discount is: " << finalPrice << endl;

return 0;

## Results

The program was tested with various scenarios to ensure its reliability. Here are the results from the test cases:

1. Test Case 1:
   1. Input: Product Price = $100, Discount Type = Percentage, Discount = 20%
   2. Output: Final Price = $80.00
2. Test Case 2:
   1. Input: Product Price = $150, Discount Type = Fixed Amount, Discount = $30
   2. Output: Final Price = $120.00
3. Test Case 3:
   1. Input: Product Price = $50, Discount Type = Invalid Discount
   2. Output: "Invalid discount type chosen!"

From the test results, we can conclude that the program works as intended for both percentage-based and fixed-amount discounts. The error handling mechanism is also effective, prompting users with an error message when an invalid discount type is entered.

1. References
   1. *C++ Programming Language* by Bjarne Stroustrup
   2. *Beginning C++ Through Game Programming* by Michael

Dawson

* 1. C++.' *The Complete Reference* by Herbert Schildt
  2. Online documentation and tutorials from cplusplus.com
  3. *Practical* C++ *Programming* by Steve Oualline

**Capital University of Science & Technology, Islamabad**

**Electrical and Computer Engineering Department**

**LAB PROJECT ASSESSMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | **Project Title** |  | | |
| 2 | **Lab** | CYG1611- Applications of Information and Communication Technologies Lab | **Semester** | Fall 2024 |
| 3 | **Student Name & Registration No.** | Student 1 | Student 2 |  |
|  |  |  |
| 4 | **Instructor Name**  **& Signature** | Mr. SM Waqas Ayub Shah | | |

**Project Demonstration**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Criteria** | **Very Poor**  **0-1** | **Poor**  **2-3** | **Satisfactory**  **4-5** | **Good**  **6-8** | **Excellent**  **9-10** | Score **Student 1** | Score **Student 2** |
| **Design Evaluation and Testing** | No or very poor design prototype and demonstration. | Design prototype is not working and no testing of design has been done | Design prototype is partially functional and little testing of design has been done. | Design prototype is functional and some testing of design has been done. | Design prototype is fully functional and design has been exhaustively tested. |  |  |
| **Usage of software tools (Visual Studio, MS Office Applications) in design and evaluation** | No or very poor software tool (Visual Studio, MS Office Applications) usage in project design and results evaluation | Insignificant evidence of software tool (Visual Studio, MS Office Applications) usage in project design and results evaluation | Little evidence of ability to select appropriate software tools (Visual Studio, MS Office Applications), in project design and results evaluation | Some evidence of skills to use software tools (Visual Studio, MS Office Applications) in project design and results evaluation | Clear evidence of skills to use software tools (Visual Studio, MS Office Applications) in project design and results evaluation |  |  |

**Project Report**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Criteria** | **Very Poor**  **0-1** | **Poor**  **2** | **Satisfactory**  **3** | **Good**  **4** | **Excellent**  **5** | Score **Student** 1 | Score **Student** 2 |
| **Literature Survey,**  **Problem Analysis and Design Procedure** | No or very poor literature survey done. No problem analysis performed. No worthwhile design procedure exists. | Insufficient literature survey Problem analysis part is skipped or does not contribute to creating an effective design. Does not follow any design procedure. | Partial literature survey. Problem Analyses performed is haphazard and design parameter selection is spontaneous. Little use of design procedure. | Adequate literature survey. Problem analysis performed correctly. Project demonstrates some use of design process. | Clear and complete literature survey, effective problem analyses is performed to choose design parameters. Project demonstrates effective use of design process. |  |  |
| **Language, Grammar and References** | A lot of spelling and grammatical mistakes with poor English. The list of references is clearly inadequate. Table of content missing. | Frequent spellings and grammatical errors. The list of references should be expanded. | Occasional spellings and grammatical errors. The list of references appears reasonable but citation does not follow standard format. | Very few spellings and grammatical errors.  Organization is good.  The list of references appears reasonable and citation follow standard format. | Almost no spelling or grammatical mistake.  Excellent organization. A comprehensive list of references is cited using the standard format. |  |  |

**Viva Voce**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Assessment Criteria** | **Very Poor**  **0-1** | **Poor**  **2** | **Satisfactory**  **3** | **Good**  **4** | **Excellent**  **5** | Score Student  1 | Score Student 2 |
| **Knowledge of Project Implementation details (Q/A)** | No or very poor knowledge of implementation and design process. | Poor knowledge of implementation and design with wrong/no answers | Satisfactory knowledge of implementation, vague answers | Adequate knowledge of project implementation with majority of correct answers | Exceptional knowledge of implementation and overall design with clear and spontaneous answers. |  |  |