

|  |  |
| --- | --- |
| Studenta Name | Afzal Ali Ahmed, Hasham Munir, Ayesha Batool |
| Students Roll no | **2024-SE-04,32,19** |
| Semester | **2nd** |
| Instructor Name | **Sir Awais** |
| Project | **Online Shopping System** |
| Course Code | **CS-1206** |
| Credit Hours | **3+1** |
| Date | **27-sep-2025** |
| Department of Software Engineering | |

**Proposal for Shopping System using OOP in C++**

**Title**

**Shopping System using Object-Oriented Programming in C++**

**Abstract**

This project is a console-based **Shopping System** developed using **Object-Oriented Programming (OOP) concepts in C++**. The system allows users to interact in two roles: **Admin** and **Customer**. Admins can manage products by adding, deleting, and updating stock, while customers can browse products, add them to a cart or wishlist, and proceed to checkout. The project also supports **file handling** for saving and retrieving product data, as well as **report generation** for stock and sales. It demonstrates the practical application of the four fundamental pillars of OOP — **Encapsulation, Inheritance, Polymorphism, and Abstraction** — in a real-world scenario.

**Objectives**

* To design and implement a shopping system using **OOP principles**.
* To allow **role-based access** for Admins and Customers.
* To manage **products, carts, and wishlists** efficiently.
* To apply **file handling** for data persistence.
* To demonstrate **exception handling** for input validation and error management.
* To practice using **static members, virtual functions, and encapsulated class design** in a real project.

**System Features**

**Admin Features**

* Add, remove, and update products.
* Manage stock levels.
* Apply discounts to products.
* Generate reports (stock report, sales report, low stock alert).

**Customer Features**

* Browse and search products by name or category.
* Add or remove items from cart.
* Maintain a wishlist.
* Checkout with total bill calculation.

**General Features**

* **Persistent Storage** using files (products.txt).
* **Input validation** with cin.clear() and cin.ignore() to avoid errors.
* **Formatted output** using iomanip for better readability.
* **Error handling** with exceptions (runtime\_error, invalid\_argument).

**Methodology**

**Object-Oriented Concepts Applied**

1. **Classes and Objects** – Used to represent real entities like Product, Cart, User, Admin, and Customer.
2. **Encapsulation** – Product details and user data were kept private, accessible only through specific functions.
3. **Inheritance** – Admin and Customer were derived from the User class.
4. **Polymorphism** – Virtual functions allowed different menu implementations for Admin and Customer.
5. **Abstraction** – Complex operations like file saving/loading were hidden inside simple functions.
6. **Static Members** – Used to count total users and products.

**Additional C++ Concepts Used**

* **File Handling:** ifstream and ofstream for saving and loading product data.
* **Exception Handling:** try–catch blocks with standard exceptions.
* **Preprocessor Directive:** #pragma once used in header files to prevent multiple inclusions.
* **Input Handling:** cin.clear() and cin.ignore() to manage invalid inputs.

**Expected Outcomes**

* A working **console-based shopping system** using OOP in C++.
* Clear demonstration of **all four OOP pillars**.
* Practical understanding of **file handling, input validation, and exception handling**.
* Ability to generate **reports** and maintain product records.
* Enhanced programming skills through modular and reusable code design.

**What We Used in OOP**

This project made full use of the **four pillars of Object-Oriented Programming**:

* **Encapsulation:** Data hiding through private variables and controlled access with public functions.
* **Inheritance:** The Admin and Customer classes inherited from the User class to reuse code.
* **Polymorphism:** Virtual functions like menu() were overridden for Admin and Customer roles.
* **Abstraction:** File operations, product management, and calculations were abstracted into functions.

Additionally, **static members** were used to keep track of overall users and product count.

**Conclusion**

The **Shopping System** project successfully demonstrates the practical application of **Object-Oriented Programming concepts** in C++. By combining encapsulation, inheritance, polymorphism, and abstraction, the system was developed in a modular and maintainable way. Features like file handling, exception handling, and input validation made the program more realistic and user-friendly. This project not only strengthened the understanding of OOP but also provided hands-on experience in building a real-world style application.