# SUPER MARKET BILLING SYSTEM

A project by:

**Lavanya Sri Chava – 2312069**
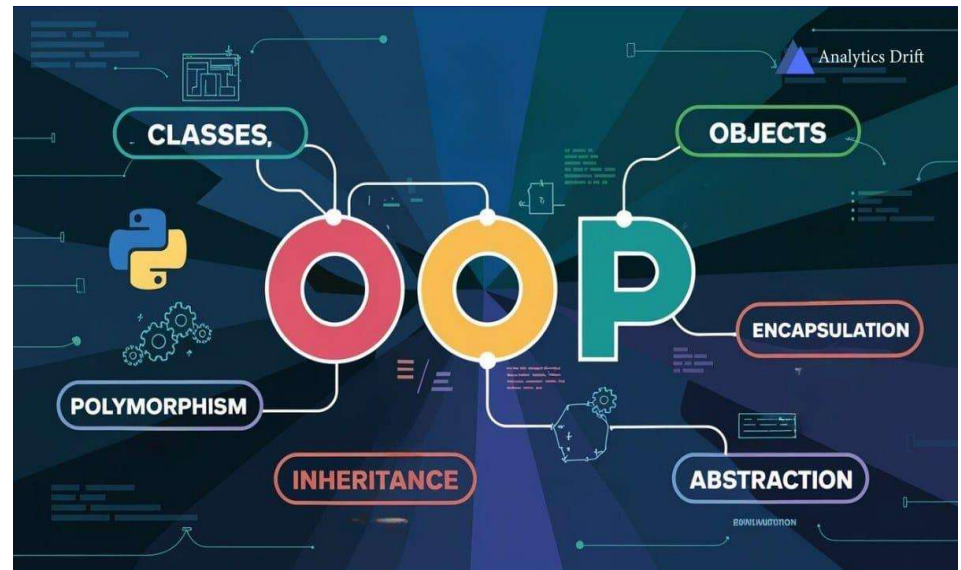
**Divya kumari- 2312077**

Department of Computer Science and Engineering

National Institute of Technology ,Silchar

# PROJECT OUTLINE :

→ Introduction

→ Problem Statement

→ Features

→ How OOPS helped?

→ Flow Chart

→ Functionality

→ Final consideration

# INTRODUCTION

Objective:

**To design and implement a supermarket billing system using C++ programming language**.

## Project Overview:

**The project aims to create a user-friendly interface for user login ,adding items to the cart ,generating bills with discounts and discounts ,providing various payment methods for customers.**

**It provides functionalities like adding items to the cart ,selecting categories to display the stock and printing receipt.**

Developed as a part of the CS 212 Object-Oriented Programming(OOP) course.

# PROBLEM STATEMENT

**Background:**

Nowadays, supermarkets are an important part of our daily lives because they make shopping easy and convenient. But for the people who work there, keeping track of items, handling payments, and making correct bills can be a tough job. To make things easier and faster, there should be a proper Supermarket Billing System. This system can help the staff do their work more smoothly and also make customers happier by saving time and avoiding mistakes.

**Problem Description :**

You are tasked with designing and implementing a supermarket billing system using C++ programming language. The system should provide functionalities to display stock ,process customer transactions, and generate bills with itemized details.

# FEATURES

**User-Friendly Interface:**
Clear and concise menu options for easy navigation, Visual feedback during item addition and bill generation for enhanced user experience.

**Scalability:**
Designed to accommodate future expansions or modifications in functionality. Flexible code structure allowing easy integration of new features.

**Enhanced User Experience:**
Incorporation of visual elements and sleep delays for a more interactive and engaging user experience and billing operations.

**Customization Options:**
Ability to customize bills with cashier ID, bill number, and timestamp for record-keeping and payment can be done based on user preference.

**Error Handling:**
Informative error messages for out-of-stock items or invalid user inputs .Robust file handling to prevent data loss or corruption.

**Billing System :**
Add items to the cart with the desired quantity ,calculate the total bill amount, including discounts and taxes.
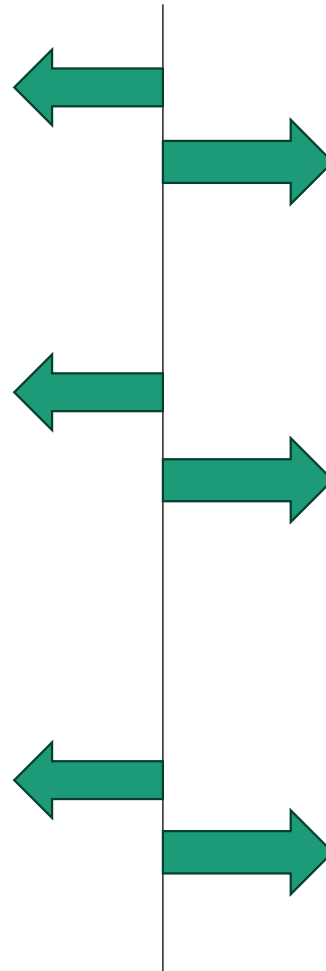
# IMPORTANT ROLE OF OOPS

## Classes

Each class represents a distinct entity with its own set of properties and behaviours Objects of these classes are created and manipulated throughout the program

## Encapsulation

implemented by using access specifiers (public, private, protected) to control the access to class members.

## Abstraction

Abstraction is achieved by hiding the internal implementation details and exposing only the essential features of an object.

## Inheritance

Inheritance is utilized to create a specialized class (Bill) from a base class (ItemDetails) to inherit common attributes and behaviours.

In my code, the Product class defines the properties and behavior of a product, while the Supermarket class defines the properties and behavior of a supermarket.

In my code, the Product class encapsulates the data members name, price, quantity, barcode, and category, as well as methods that operate on that data, such as getName() and getPrice().

In my code, the Product class abstracts the implementation details of a product, such as how the price is calculated, and only exposes the necessary information, such as the product name and price

Inheritance is not explicitly used in your code. However, if you were to create a new class, such as Discounted Product, that inherits from the Product class, you could reuse the properties and behavior of the Product class and add new properties and behavior specific to discounted products.

## Constructor Overloading

the concept of having more than one constructor with different parameters so that every constructor can perform a different task.

## Destructor

a member function that is invoked automatically when the object goes out of scope or is explicitly destroyed by a call to delete or delete[]

## File Handling (I/O)

File handling is utilised to read and write data from/to external files, such as maintaining the inventory (shop.txt) of the super market.

In my code, the Product class has a constructor that takes six parameters: name price, quantity, barcode, category, and bogooffer. This is an example of constructor overloading, where a single constructor can be used to initialize objects with different parameters.

My code does not explicitly define a destructor for the Product or Supermarket classes. However, the compiler will generate a default destructor for each class, which will automatically release any dynamically allocated memory.
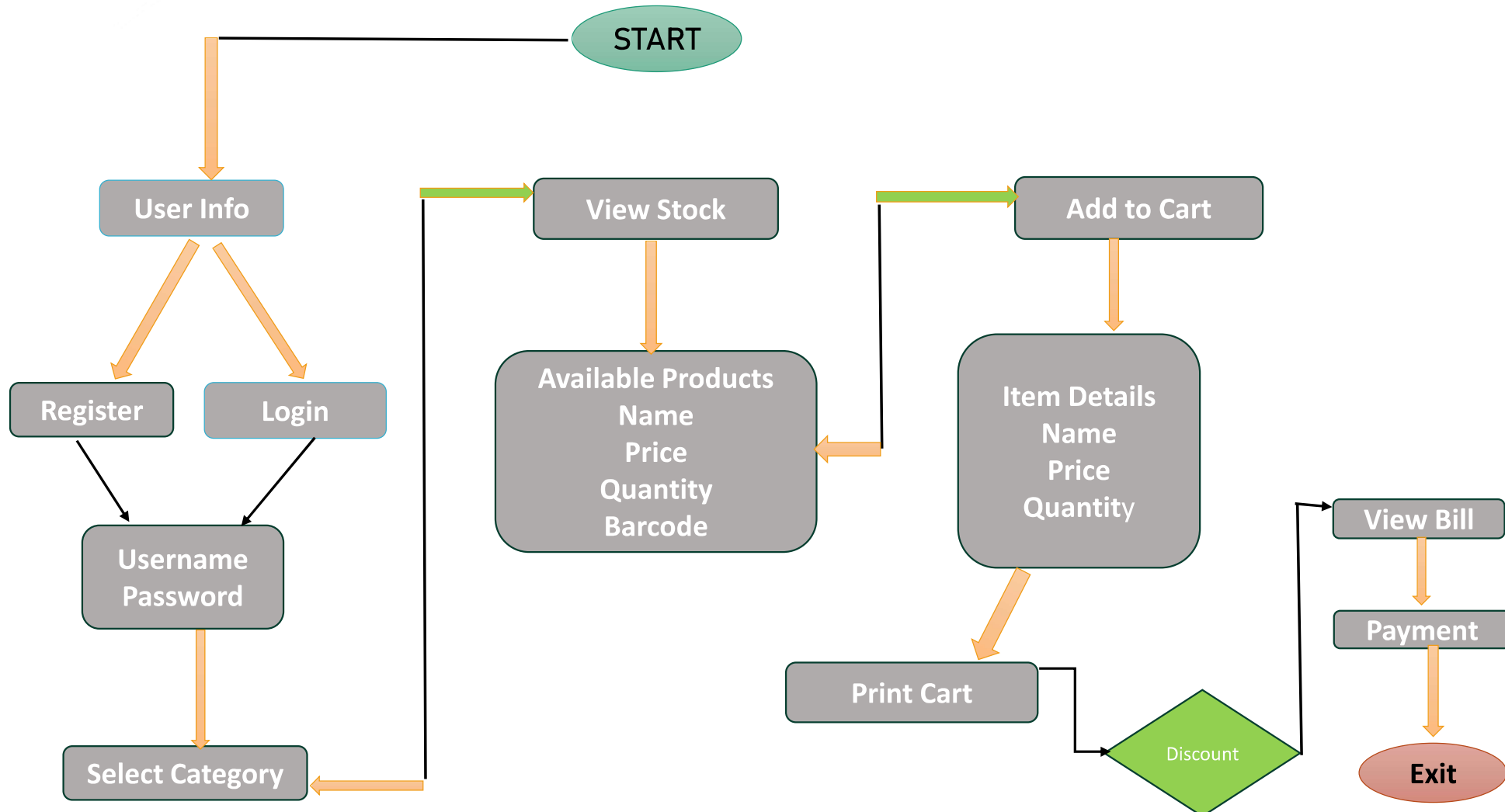
In my code, file handling is used to generate receipts and save them to a file. The generateReceipt function uses an ofstream object to create an output file stream, and the << operator is used to write data to the file.

# FLOW CHART

**START**

**User Info**

**Register**

**Login**

**Username
Password**

**Select Category**

**View Stock**

**Available Products
Name
Price
Quantity
Barcode**

**Add to Cart**

**Item Details
Name
Price
Quantity**

**Print Cart**

Discount

**View Bill**

**Payment**

**Exit**

# FUNCTIONALITY OF PROJECT

Output                                                    Clear

```
1. Register
2. Login
Choose option: I
```

**MAIN MENU**

```
1. Register
2. Login
Choose option: 1
Enter username: lavanya
Enter password: 123456
Registration successful!
```

```
Select Category:
1. Fruits
2. Snacks
3. Juices
4. Accessories
5. Clothes
6. All
Enter your choice: 6
```

```
All Available Products:
Product: Apple | Category: Fruits | Price: ₹10.50 | Quantity: 100 | Barcode: 1001
Product: Banana | Category: Fruits | Price: ₹10.00 | Quantity: 100 | Barcode: 1002
Product: Orange | Category: Fruits | Price: ₹10.50 | Quantity: 100 | Barcode: 1003
Product: Grapes | Category: Fruits | Price: ₹10.50 | Quantity: 100 | Barcode: 1004
Product: Pineapple | Category: Fruits | Price: ₹10.50 | Quantity: 100 | Barcode: 1005
Product: Milk | Category: Juices | Price: ₹20.00 | Quantity: 30 | Barcode: 3001 | Offer: Buy One Get One Free!
Product: sprite | Category: Juices | Price: ₹20.00 | Quantity: 30 | Barcode: 3002 | Offer: Buy One Get One Free!
Product: Thumsup | Category: Juices | Price: ₹20.00 | Quantity: 30 | Barcode: 3003 | Offer: Buy One Get One Free!
Product: Frootie | Category: Juices | Price: ₹20.00 | Quantity: 30 | Barcode: 3004 | Offer: Buy One Get One Free!
Product: Paperboat | Category: Juices | Price: ₹20.00 | Quantity: 30 | Barcode: 3005 | Offer: Buy One Get One Free!
Product: Chain | Category: Accessories | Price: ₹10.00 | Quantity: 50 | Barcode: 4001
Product: Chowkar | Category: Accessories | Price: ₹10.00 | Quantity: 50 | Barcode: 4002
Product: Bangles | Category: Accessories | Price: ₹10.00 | Quantity: 50 | Barcode: 4003
Product: Nosering | Category: Accessories | Price: ₹10.00 | Quantity: 50 | Barcode: 4004
Product: Track pant | Category: Clothes | Price: ₹500.00 | Quantity: 20 | Barcode: 5001
Product: Trousers | Category: Clothes | Price: ₹500.00 | Quantity: 20 | Barcode: 5002
Product: Jeans | Category: Clothes | Price: ₹500.00 | Quantity: 20 | Barcode: 5003
Product: Skirt | Category: Clothes | Price: ₹500.00 | Quantity: 20 | Barcode: 5004
Product: T-shirt | Category: Clothes | Price: ₹500.00 | Quantity: 20 | Barcode: 5005
Product: Chips | Category: Snacks | Price: ₹15.00 | Quantity: 50 | Barcode: 2002
Product: Biscuits | Category: Snacks | Price: ₹30.00 | Quantity: 40 | Barcode: 2001
Product: Madangles | Category: Snacks | Price: ₹30.00 | Quantity: 40 | Barcode: 2004
Product: Moongdal | Category: Snacks | Price: ₹30.00 | Quantity: 40 | Barcode: 2005
```

```
10 x Chips added to the cart.
1 x Chain added to the cart.
2 x Track pant added to the cart.
4 x Moongdal added to the cart.
Special Offer Applied: Buy One Get One Free on sprite!
10 x sprite added to the cart.
```

```
Receipt has been saved as 'receipt_1041.txt'.

Select Payment Method:
1. Cash
2. Credit/Debit Card
3. Net Banking
4. UPI (Paytm, Google Pay, PhonePe)
Enter choice (1-4):
```

TOTAL BILL OF SELECTED ITEMS

```
=========================================
        SUPERMARKET BILLING SYSTEM
=========================================

Receipt Number: 1041
Date: 2025-4-8 21:37:52
-----------------------------------------

Item      Qty      Price     Total
-----------------------------------------

Chips     10       ₹15       ₹150
Chain     1        ₹10       ₹10
Track pant         2         ₹500      ₹1000
Moongdal           4         ₹30       ₹120
sprite    10       ₹20       ₹200

-----------------------------------------
Subtotal: ₹1480
Discount: ₹148
GST (18%): ₹239.76
Final Amount: ₹1571.76
=========================================
```

# FINAL CONSIDERATION

OUR TEAM MEMBERS

THANK YOU

**LAVANYA SRI CHAVA**
**2312069**

**DIVYA KUMARI**
**2312077**

CONTACT US:
9985175689
9508940077