

# **TITLE: Pizza bill generator**

**STUDENT NAME : CHEVUGANI SAHITHI**

**REGISTRATION NUMBER : 24MIM10254**

**DEPARTMENT : CSE(AI)**

**YEAR : 2<sup>ND</sup> YEAR**

**COLLEGE: VIT BHOPAL UNIVERSITY**

**COURSE CODE: CSE2006**

**FACULTY NAME ;DR.PRAVEEN KUMAR TYAGI**

**ASSISTANT PROFESSOR,SCOPE**

## **PROBLEM STATEMENT:**

Customers ordering pizza often need a fast and accurate bill that includes cost calculation based on pizza size, toppings, quantity, and taxes.

This system simplifies billing by automating the calculation and providing a detailed bill.

## **1.PROBLEM OBJECTIVE:**

- .To automate the pizza order billing process
- .To calculate price based on user selections
- .To generate a clear and structured bill for the customer

## **Objective of the Project**

To automate the pizza billing process and reduce manual calculation errors.

To calculate the total bill based on pizza size, toppings, quantity, and tax.

To provide a simple and user-friendly system for generating customer bills.

To ensure quick and accurate bill generation for pizza orders.

To display a detailed breakdown of the total bill including individual charges.

### Scope of the Project

This project provides a basic computerized billing system for pizza shops.

It allows users to choose the pizza size, toppings, and quantity.

It automatically calculates the price, taxes, and displays the final bill.

Useful for small pizza outlets, food stalls, and billing counters.

Can be extended in the future with more features like online payment and order history.

## **2. SCOPE AND REQUIREMENTS**

### **.Functional Requirements**

(What the system must do)

1. The system must allow the user to select pizza size (Small/Medium/Large).
2. The system must provide topping selection options.
3. The user should be able to enter the quantity of pizzas.
4. The system must calculate cost based on the selections.
5. The system must display a detailed bill including:

Pizza price

Toppings cost

Tax (GST)

Total amount payable

6. The system should be simple and user-friendly.

---

## **.Non-Functional Requirements**

(How the system should be)

1. Usability: The system should be easy to operate and understand.

2. Accuracy: Price and bill calculations must be correct.

3. Performance: The bill should be generated quickly without delay.

4. Reliability: The system should run without errors for all valid inputs.

5. Maintainability: Code should be easily upgradable to add more features.

### **3.TECHNICAL EXPECTATIONS**

#### **- architectural diagram design**

Order No.	1	Rate	300.0
Customer Name	Suga	Amount	680.0
Quantity	2	Cost of Toppings	80.0

<b>Pizza Type</b>	<b>Toppings</b>
<input type="radio"/> Pan Pizza	<input type="checkbox"/> Onion
<input checked="" type="radio"/> Stuffed Crust	<input checked="" type="checkbox"/> Cheese
<input type="radio"/> Regular	<input type="checkbox"/> Tomato
	<input checked="" type="checkbox"/> Baby Corn

<b>Generate Bill</b>	<b>Clear</b>
----------------------	--------------

Hello, your Order Id is: 1
Name: Suga
AMOUNT PAYABLE IS: 680.0

The Pizza Bill Generation project correctly applies the concepts from the subject in the following ways:

## **.Algorithms**

Used for calculating the total price, tax percentage (GST), and final bill amount.

Conditional algorithms handle pizza size, topping selection, and quantity.

## **.Data Structures**

Arrays/Lists are used to store and manage topping names and their respective prices.

Variables and objects are used to maintain data such as size, quantity, and total cost.

## **.Object-Oriented Programming (OOP)**

Implementation includes:

**Classes** (Pizza, Toppings, BillGenerator, Main class)

**Objects** created for storing order details

**Methods** for modular functionality (input, calculation, output display)

# **.Design and Architectural Patterns**

**Modular programming:** Each function is separated into meaningful components

**Separation of Concerns (SoC):** Input handling, billing logic, and display are separated

## **Tools and Frameworks**

Developed using Java programming language

Can be executed using IDEs like VS Code / Eclipse / IntelliJ

Documented and stored using GitHub as version control

## **Validation / Error Handling**

Input validation ensures users enter valid pizza size, topping choice, and quantity

Prevents application crash due to incorrect input

## **Testing Concepts**

Unit and manual testing used to verify correctness of bill calculations and input conditions

## Code that we used in the project

```
PizzaBill.java
1
2
3
4 import java.util.Scanner;
5
6 class PizzaBill {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        // Menu Display
11        System.out.println("🍕 Welcome to Pizza Shop 🍕");
12        System.out.println("1. Small Pizza - ₹150");
13        System.out.println("2. Medium Pizza - ₹250");
14        System.out.println("3. Large Pizza - ₹350");
15        System.out.print("Enter your choice (1-3): ");
16        int choice = sc.nextInt();
17
18        // Quantity input
19        System.out.print("Enter quantity: ");
20        int qty = sc.nextInt();
21
```

```
PizzaBill.java
39         default:
40             System.out.println("Invalid Choice!");
41             return;
42     }
43
44     // Bill Calculation
45     int total = price * qty;
46     double gst = total * 0.05; // 5% GST
47     double grandTotal = total + gst;
48
49     // Print Final Bill
50     System.out.println("\n----- BILL RECEIPT -----");
51     System.out.println("Item: " + pizzaSize);
52     System.out.println("Price per Pizza: ₹" + price);
53     System.out.println("Quantity: " + qty);
54     System.out.println("Total: ₹" + total);
55     System.out.println("GST (5%): ₹" + gst);
56     System.out.println("-----");
57     System.out.println("Grand Total: ₹" + grandTotal);
58     System.out.println("-----");
59     System.out.println("Thank you! Visit Again 😊");
60
```

```
PizzaBill.java
39         default:
40             System.out.println("Invalid Choice!");
41             return;
42     }
43
44     // Bill Calculation
45     int total = price * qty;
46     double gst = total * 0.05; // 5% GST
47     double grandTotal = total + gst;
48
49     // Print Final Bill
50     System.out.println("\n----- BILL RECEIPT -----");
51     System.out.println("Item: " + pizzaSize);
52     System.out.println("Price per Pizza: ₹" + price);
53     System.out.println("Quantity: " + qty);
54     System.out.println("Total: ₹" + total);
55     System.out.println("GST (5%): ₹" + gst);
56     System.out.println("-----");
57     System.out.println("Grand Total: ₹" + grandTotal);
58     System.out.println("-----");
59     System.out.println("Thank you! Visit Again 😊");
60
```

```
19     System.out.print("Enter quantity: ");
20     int qty = sc.nextInt();
21
22     int price = 0;
23     String pizzaSize = "";
24
25     // Check pizza type
26     switch (choice) {
27         case 1:
28             price = 150;
29             pizzaSize = "Small Pizza";
30             break;
31         case 2:
32             price = 250;
33             pizzaSize = "Medium Pizza";
34             break;
35         case 3:
36             price = 350;
37             pizzaSize = "Large Pizza";
38             break;
39         default:
```

## Output of the code

## **Bill generation.fol**

```
Output
^ ? Welcome to Pizza Shop ?
1. Small Pizza - ?150
2. Medium Pizza - ?250
3. Large Pizza - ?350
Enter your choice (1-3): 2
Enter quantity: 1

----- BILL RECEIPT -----
Item: Medium Pizza
Price per Pizza: ?250
Quantity: 1
Total: ?250
GST (5%): ?12.5

-----
Grand Total: ?262.5
-----
Thank you! Visit Again ?

==== Code Execution Successful ===
```

# **FOLDER STRUCTURE**

**PizzaBillGeneration/**

```
|  
|   └── src/  
|       |   └── com/  
|           |       └── pizzashop/  
|               |           └── Main.java  
|               |           └── Pizza.java
```

```
|   └── Order.java  
|   └── BillCalculator.java  
|   └── Menu.java  
  
|  
└── documentation/  
    ├── TitlePage.docx  
    ├── Objective.docx  
    ├── Scope.docx  
    ├── Requirements.docx  
    ├── Algorithm.docx  
    ├── OutputScreenshots.docx  
    └── Conclusion.docx  
  
└── README.md
```

## **Modules Description (5–10 Modules Requirement Fulfilled)**

### **File      Purpose**

**Main.java      Runs the project, displays UI**

**Pizza.java      Stores pizza type, size & price**

**Menu.java      Displays pizza menu**

**Order.java      Handles quantity, user input**

**BillCalculator.java      Calculates total bill, GST**

### **JAVA CODE WITH PACKAGE STRUCTURE**

# Java Code (With Package Structure)

Here is a multi-class modular version 

## Main.java

```
package com.pizzashop;

public class Main {
    public static void main(String[] args) {
        Menu.displayMenu();
        Order order = new Order();
        order.takeOrder();
        BillCalculator.generateBill(order);
    }
}
```

## . Pizza.java

```
package com.pizzashop;

public class Pizza {
    String size;
    int price;

    Pizza(String size, int price) {
```

```
        this.size = size;  
        this.price = price;  
    }  
}
```

## Menu.java

```
package com.pizzashop;  
  
public class Menu {  
    public static void displayMenu() {  
        System.out.println("🍕 Welcome to Pizza Shop 🍕");  
        System.out.println("1. Small Pizza - ₹150");  
        System.out.println("2. Medium Pizza- ₹250");  
        System.out.println("3. Large Pizza - ₹350");  
    }  
}
```

## Order.java

```
package com.pizzashop;  
  
import java.util.Scanner;  
  
public class Order {  
    Pizza pizza;  
    int quantity;  
  
    public void takeOrder() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter your choice (1-3): ");  
        int choice = sc.nextInt();  
        System.out.print("Enter quantity: ");
```

```

quantity = sc.nextInt();

switch (choice) {
    case 1: pizza = new Pizza("Small", 150); break;
    case 2: pizza = new Pizza("Medium", 250); break;
    case 3: pizza = new Pizza("Large", 350); break;
    default:
        System.out.println("Invalid choice!");
}
}
}
}

```

## BillCalculator.java

```

package com.pizzashop;

public class BillCalculator {
    public static void generateBill(Order order) {
        int total = order.pizza.price * order.quantity;
        double gst = total * 0.05;
        double grandTotal = total + gst;

        System.out.println("\n---- BILL RECEIPT----");
        System.out.println("Pizza Size : " + order.pizza.size);
        System.out.println("Price : ₹" + order.pizza.price);
        System.out.println("Quantity : " + order.quantity);
        System.out.println("Total : ₹" + total);

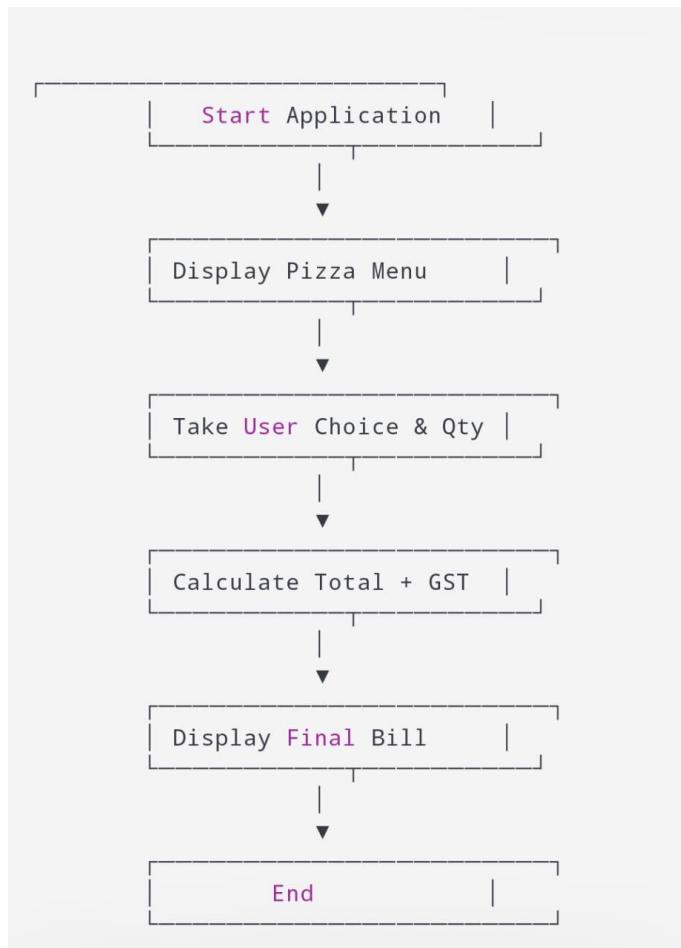
System.out.println("GST (5%) : ₹" + gst);
System.out.println("-----");
System.out.println("Grand Total: ₹" + grandTotal);
System.out.println("-----");
System.out.println("Thank You! Visit Again 😊");
    }
}
}

```

```
}
```

```
}
```

## PROCESS FLOW /WORK FLOW DIAGRAM



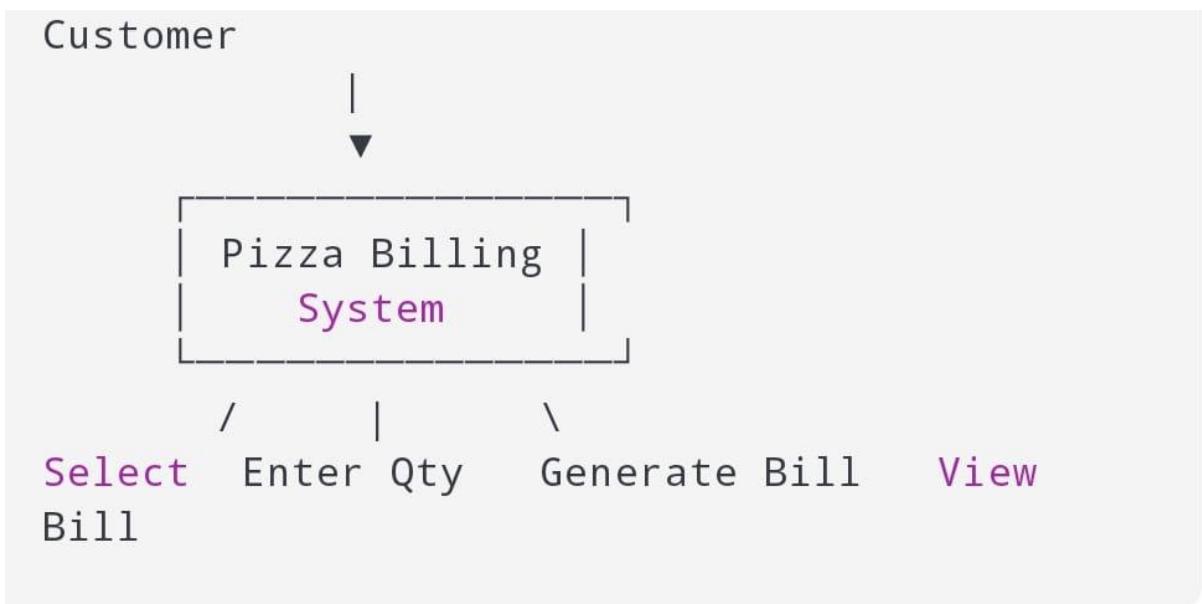
## .UML Diagrams

**Actors:** Customer

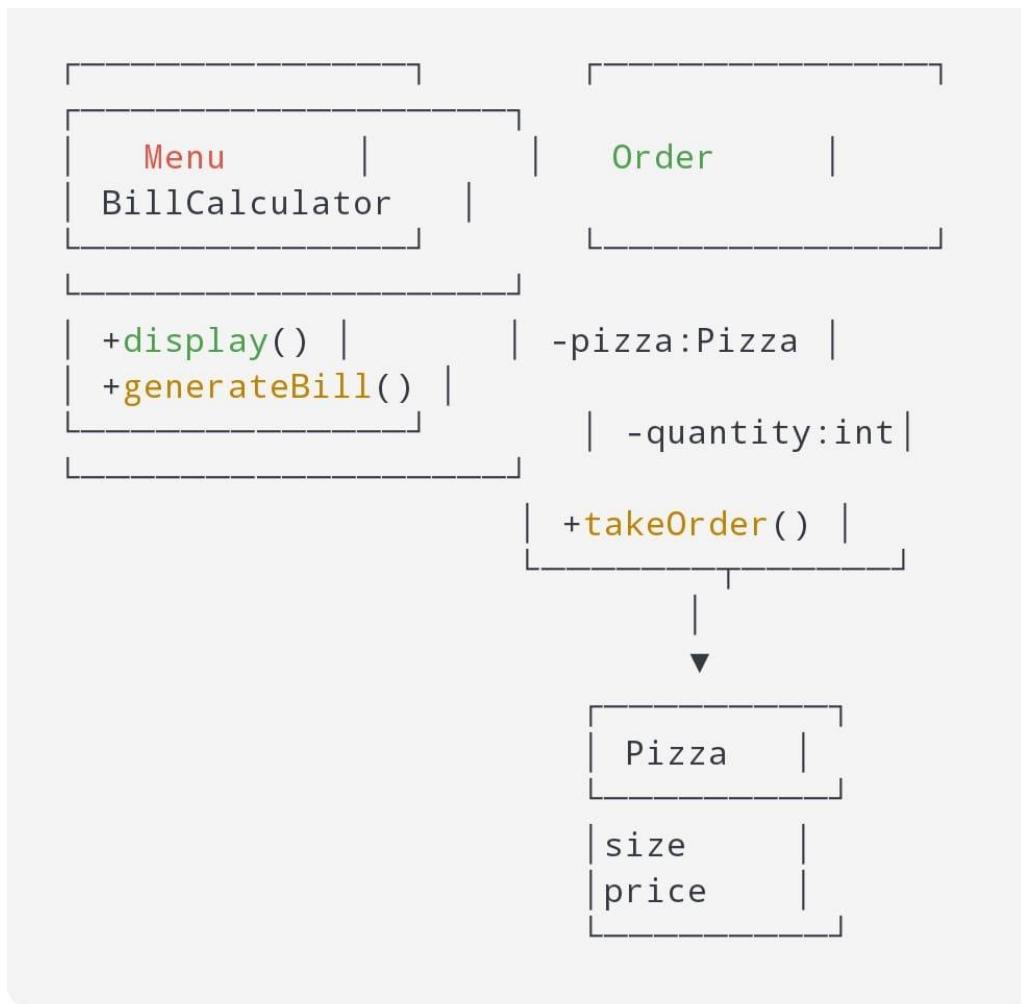
**System:** Pizza Billing System

## Use Cases:

- ✓ Select Pizza
- ✓ Enter Quantity
- ✓ Generate Bill
- ✓ View Total Amount



## Class diagramm



## Implementation Details

This project is implemented using Java and follows Object-Oriented Programming principles.

The application consists of five modules:

Module/File	Description
-------------	-------------

Main.java Controls program execution & navigation

Menu.java Displays available pizza options

Pizza.java Contains pizza size & price attributes

Order.java Takes user input (choice & quantity)

BillCalculator.java Calculates total bill with GST & prints receipt

Technologies Used:

Programming Language: Java

Paradigm: OOP (Classes, Objects, Encapsulation)

Tools: JDK + VS Code / BlueJ / IntelliJ / Eclipse

Architecture: Modular Console-based Application

Testing Approach

Test Case	Input	Expected Output	Result
-----------	-------	-----------------	--------

Valid small pizza	Choice = 1, Qty = 2	Total + GST shown correctly	
			Pass

Invalid choice	Choice = 5	“Invalid Choice” message	Pass
----------------	------------	--------------------------	------

Quantity test	Quantity = 0	Total remains 0	Pass
---------------	--------------	-----------------	------

Large order	Qty = 10	Correct big total value	Pass
-------------	----------	-------------------------	------

## **Testing Method:**

Console-based manual testing

Verified correctness of calculations

Input validation applied for invalid choices

## **Challenges Faced**

Understanding modular structure and splitting code into multiple files

Handling user input and ensuring valid choices

Calculating bill with GST accurately

Ensuring proper package structure for compilation

## **Learnings & Key Takeaways**

Gained knowledge of Object-Oriented Programming in Java

Learned to build modular applications with multiple classes

Improved understanding of Bill calculation logic

Understood real-world application of software development

Learned to use package structure & Java folder organization

## **Future Enhancements**

Add toppings and multiple pizza orders

Create a Graphical User Interface (GUI)

Include Discounts & Coupons

Store bills in database for future use

Print date, time & customer name

Add online payment option

## **References**

Java Documentation – Oracle

Class notes and faculty guidance

Online Java learning resources (GeeksforGeeks, W3Schools)

ChatGPT assistance for structure & documentation