



Lab Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	40
Understanding/Analysis	10					
Implementation	15					
Accuracy	10					
Task Efficiency	5					
Total obtained mark						
Comments						

Semester: Spring 2025

Student Name: Md. Sohag Milon Shanto

Student ID:242-35-405

Batch: 43 SWE

Section:43-C

**Course Code: SE-122
Lab**

Course Name: Structured Programming

Course Teacher Name: Suprove Chandra Sarker

Designation: Lecturer

Submission Date: ...08.... /...04.../...25....



LAB REPORT: RESTAURANT MANAGEMENT SYSTEM IN C

Where you can order food, see
your order and make the payment

Created By Team Collaboration—

1: Aniruddho Saha(786)

2: Samia Islam Lima(714)

3: Md. Sohag Milon Shanto(405)

Abstract

This is a command-line Restaurant Management System written in the C programming language. It enables users to interact with a menu-driven system to place orders, see what is available on the menu, and see past orders. The system employs structures to store menu and order information, loops for repeated input, and arrays to handle numerous items. This project demonstrates real-world application of C programming concepts like variables, functions, structures, control statements, and user input/output.

Objective

The purpose of this lab project is to create a small, efficient system for restaurant order processing utilizing systematic programming techniques in C. The goal is to incorporate an understanding of programming basics such as control structures, arrays, functions, and user-defined data types (structs) into an application-based simulation scenario in real life.

Tools and Technologies Used

- Programming Language: C
- Compiler: Code ::Blocks
- OS: Windows

C Functionalities Used in the Project

Header Files

- `#include <stdio.h>`: For input and output operations.
- `#include <stdlib.h>`: For general functions like `exit()`.
- `#include <string.h>`: For string manipulation functions like `strcpy()`, `strncpy()`.

Variables and Data Types

- `int`: For indexing menu items, order ID, quantity, counters, and user input.
- `float`: For prices of items and computation of total order values.
- `char`: Used to store character inputs like payment method.
- `char array`: Stored item names and user input.

Constants and Macros

- `#define MAX_ITEMS, MAX_ORDERS, etc.`: Stored array size boundaries.

Structures (struct)

- `MenuItem`: Stores `itemName` and `price`.
- `Order`: Stores `orderId`, order list of items, quantity, total amount, and payment method.

Arrays

- Menu and Orders are stored using structured arrays.
- Character arrays for item names and input.

Loops

- for: To traverse the menu and orders.
- while: For loop while user exits.

Functions

- displayMenu(): Display all menu items and prices.
- takeOrder(): Manage item selection, quantity, and payment mode.
- viewOrders(): Display all previous orders.
- main(): Offer main menu and user navigation.

Input/Output

- `printf()`, `scanf()`, `fgets()`: For printing messages and input.

Implementation Overview

1. Menu Display: Predefined list of items is printed by display Menu().
2. Taking Orders: Number of items and quantity are taken from user as input.
3. Payment Method: Cash ('C') or Online ('O') is selected by the user.
4. View Orders: Stored past orders are displayed.

```

#define MAX_ITEM_NAME 20
#define MAX_ORDER_ITEMS 10

typedef struct {
    char itemName[MAX_ITEM_NAME];
    float price; // Price is 3.99
} MenuItem;

typedef struct {
    int orderId;
    char items[MAX_ORDER_ITEMS][MAX_ITEM_NAME];
    int quantity[MAX_ORDER_ITEMS];
    int itemCount;
    float totalAmount; // Total amount is 33.95
    char paymentMethod;
} Order;

MenuItem menu[MAX_ITEMS] = {
    {"Burger", 8.9},
    {"Sandwich", 6.0},
    {"Hot Dog", 1.0},
    {"Pasta", 10.0},
    {"Waffle", 7.5},
    {"Coke", 3.99},
    {"Drinks", 3.0}
};

Order orders[MAX_ORDERS];
int orderCount = 0;

void displayMenu() {
    printf("\n--- Menu ---\n");
    for (int i = 0; i < MAX_ITEMS; i++) {
        printf("%d. %-15s: %.2f\n", i + 1, menu[i].itemName, menu[i].price);
    }
    printf("\n");
}

int takeOrder() {
    int orderCount = 0;
    while (orderCount < MAX_ORDER_ITEMS) {
        printf("Menu Item %d: ", orderCount);
        return;
    }
}

```

Inputs and Outputs

Inputs:

- Number of menu item (int)
- Quantity (int)
- Payment Method (char: C or O)

```
--- Restaurant Management System ---
1. View Menu
2. Place Order
3. View Orders
4. Exit
```

Outputs:

- Printed menu
- Order summary with total cost
- Order list saved
- Thank you note and Order ID

```
--- Restaurant Menu (Prices in Taka) ---
1. Burger      : 80.00 Taka
2. Sandwich    : 60.00 Taka
3. Hot Dog     : 110.00 Taka
4. Pasta       : 100.00 Taka
5. Momo        : 70.00 Taka
6. Shawarma    : 90.00 Taka
7. Drinks      : 30.00 Taka

Enter item number (0 to finish): 2
Enter quantity: 1
Enter item number (0 to finish): 7
Enter quantity: 1
Enter item number (0 to finish): 0
Total Amount: 90.00 Taka
Payment Method (C for Cash, 0 for Online): C
Order placed successfully. Order ID: 1
Thank you for your order!
```

Conclusion

This project effectively employs a simple Restaurant Management System employing structured programming in C. It effectively integrates simple programming concepts such as functions, loops, arrays, and user-defined structures. Interactive functionality enables users to simulate real-time ordering and payment, and thus it is an effective learning project for beginners to learn C programming fundamentals. The modularity ensures readability, usability, and extensibility in the future for extensions.

END OF REPORT
