NED University of Engineering and Technology

Department of Computer Science & Information Technology

Complex Computing Project (CCP)

Programming Fundamentals CT-175

Instructor: Mr. Muhammad Abdullah

Group Members:

Laiba Qureshi CT-25154
Fatima Nasir CT-25155
Rateeba Ahmed CT-25173

Project Title: Automated Drink Ordering and Billing System

Abstract

This project, Café Billing System Using Arrays and Switch Statements, is a simple C program that simulates a café's ordering and billing process. It allows the user to select a drink, choose its size, and add an optional sweet item. The program uses arrays to store menu items and their corresponding prices, making it easier to manage and modify the menu. It also employs switch statements to handle user selections efficiently. The program calculates the total bill, including a 10% tax, and displays a clear order summary. This project demonstrates the use of arrays, conditional structures, and arithmetic operations in C programming while reinforcing the concept of user interaction through console input and output.

INDEX

1. Project Description

The Café Billing System is a console-based application developed in C programming language, designed to simulate the real-world operations of a café, including order placement and bill generation. The system allows customers to select from a variety of drinks, choose the preferred size, and add optional sweet items, all through an interactive menu. Arrays are used to store menu items and their respective prices, which simplifies the management of items and enables easy updates or expansions to the menu. By incorporating conditional statements and arithmetic operations, the program calculates the subtotal, applies a fixed tax rate, and provides the final total bill. The system ensures accurate price computation, reduces human errors that occur in manual billing, and generates a clear and organized order summary for the user. This project not only demonstrates the practical application of fundamental C programming concepts such as arrays, loops, and switch statements but also provides a foundation for more advanced features, such as multi-order processing, data storage, receipt printing, and user interface enhancements. It serves as a practical example of how basic programming constructs can be applied to solve everyday commercial problems efficiently while offering potential for further development into a comprehensive café management system.

2. Project Methodology

2.1 Dataset / Input Data

- Drink Names (Spanish Latte, Iced Brown Sugar Espresso, Tiramisu Latte, Strawberry Matcha)
- Drink Prices
- Temperature of drink (Hot or Iced)
- Addition of charges if iced drink
- Size Options (Small, Medium, Large)
- Size Charges
- Sweet Add-ons (Lemon Tart, Chocolate Chip Cookie, Almond Croissant, No Sweet)
- Sweet Prices
- Tax Rate (10%)

2.2 Tools and Technologies

- **Language:** C Programming
- **IDE:** Dev C++
- **Platform:** Windows OS
- **Compiler:** GCC

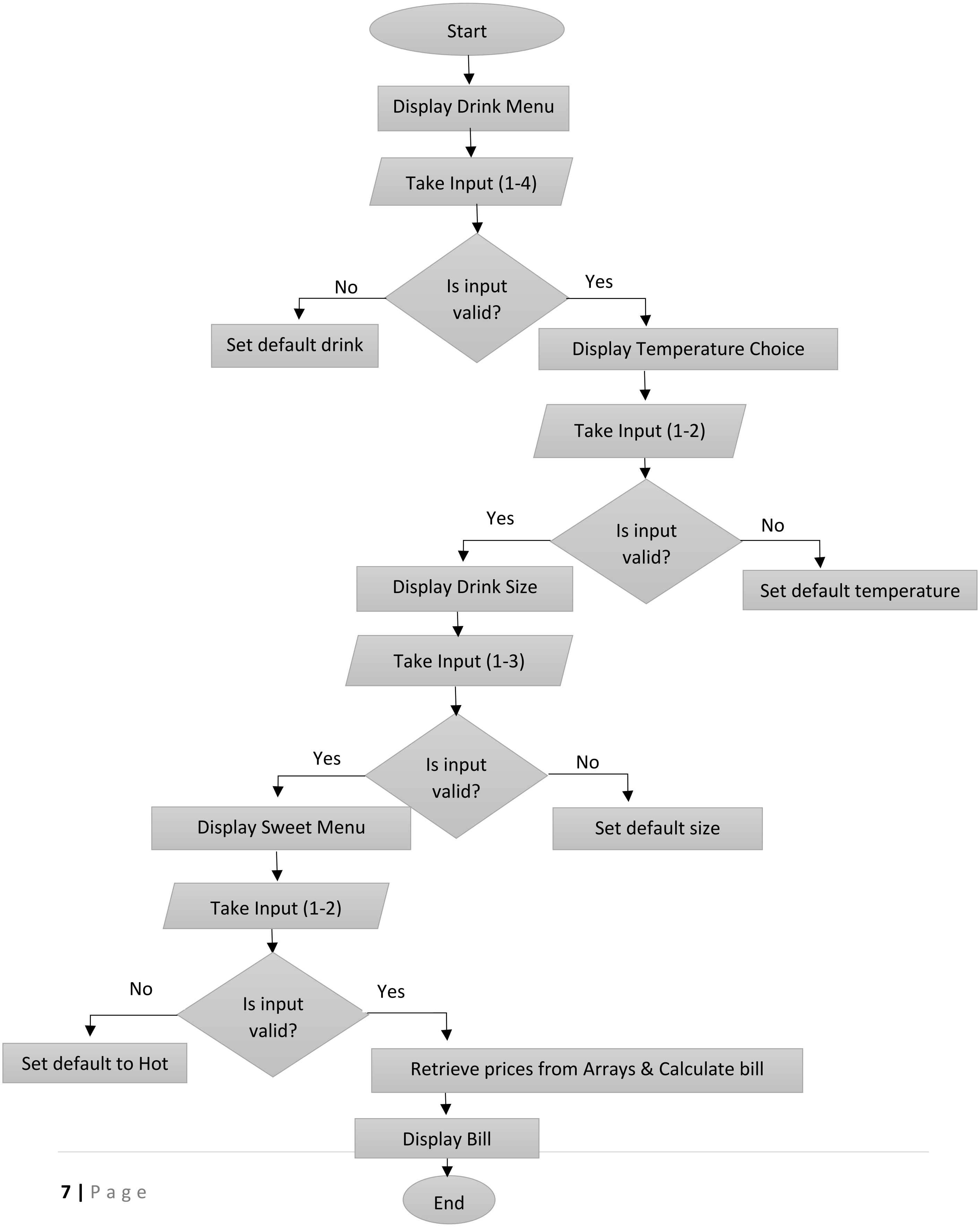
2.3 Algorithm / Pseudocode

```
1. START
2. Display Welcome message
3. Display Drink Menu
4. Input drinkChoice
5. IF invalid THEN set default drink
6. Display Temperature Option (Hot/Iced)
7. Input tempChoice
8. IF invalid THEN set Hot
9. Display Size Menu
10.Input sizeChoice
11.IF invalid THEN set Small
12.Display Sweet Menu
13.Input sweetChoice
14.IF invalid THEN set No Sweet
15.Calculate total = drinkPrice + tempPrice + sizePrice + sweetPrice
16.Calculate tax = total * TAX_RATE
17.Calculate finalTotal = total + tax
18.Display Order Summary (Drink, Temperature, Size, Sweet, Subtotal, Tax, Total)
19.Display Thank You Message
20.END
```

2.4 Objectives

- ✓ To automate café billing operations
- ✓ To implement arrays and conditional statements effectively
- ✓ To generate accurate bills including tax and itemized breakdown
- ✓ To simulate a real-world ordering process in a simple console format

2.5 Flowchart



2.6 Expected Outcomes

The expected outcome of this project is a functional **console-based Cafe Billing System** that accurately performs drink ordering and billing operations.

The system should:

- Allow users to select drinks, sizes, and sweet add-ons using numeric inputs.
- Retrieve and process menu data stored in arrays.
- Calculate total bills, including tax, without human intervention.
- Display an itemized order summary with each selected option and corresponding prices.
- Ensure accuracy and consistency in billing through automated calculations.
- Provide a simple and user-friendly interface for smooth interaction.

2.7 Goals

- To apply programming fundamentals to solve real-world problems.
- To demonstrate proficiency in **arrays, switch statements, and loops** in C.
- To create a structured and modular program for efficient order handling.
- To simulate a café environment through a logical flow of user choices and calculations.
- To encourage logical problem-solving and practical implementation of classroom concepts.

3.Code Implementation

3.1 Code

```
#include <stdio.h>
#include <string.h>


int main()
{
   int drinkChoice, sizeChoice, sweetChoice, tempChoice;
   float total, tax, finalTotal;
   float tempPrice = 0;
   const float TAX_RATE = 0.10; // 10% tax


   // Arrays for menu items and prices
   char drinks[4][40] = {"Spanish Latte", "Iced Brown Sugar Espresso", "Tiramisu
Latte", "Strawberry Matcha"};
   float drinkPrices[4] = {600, 750, 900, 750};

   char sizes[3][10] = {"Small", "Medium", "Large"};
   float sizePrices[3] = {0, 250, 350};

   char sweets[4][40] = {"Lemon Tart", "Chocolate Chip Cookie", "Almond
Croissant", "No Sweet"};
   float sweetPrices[4] = {300, 250, 400, 0};


   printf("============================\n");
   printf("    Welcome to RatFatLab Cafe\n");
   printf("============================\n\n");


   // Drink selection
   printf("Available Drinks:\n");
   for (int i = 0; i < 4; i++)
```

```
{
   printf("%d. %s (Rs. %.2f)\n", i + 1, drinks[i], drinkPrices[i]);
}

printf("Enter your drink choice (1-4): ");
scanf("%d", &drinkChoice);

if (drinkChoice < 1 || drinkChoice > 4)
{
   printf("Invalid choice! Defaulting to Spanish Latte.\n");
   drinkChoice = 1;
}


printf("\nWould you like it:\n"); // hot or cold drink
      printf("1. Hot (+Rs. 0)\n");
      printf("2. Iced (+Rs. 150)\n");
      printf("Enter your choice (1-2): ");
      scanf("%d", &tempChoice);

      if (tempChoice == 1)
      {
   tempPrice = 0;
      }
      else if (tempChoice == 2)
      {
   tempPrice = 150; // +Rs.150 added for iced drink
      }
      else
      {
         printf("Invalid choice! Defaulting to Hot.\n");
   tempChoice = 1;
   tempPrice = 0;
      }
// Size selection
printf("\nChoose Size:\n");
for (int i = 0; i < 3; i++)
```

```
        printf("%d. %s (+Rs. %.2f)\n", i + 1, sizes[i], sizePrices[i]);
    printf("Enter your choice (1-3): ");
    scanf("%d", &sizeChoice);

    if (sizeChoice < 1 || sizeChoice > 3)
    {
        printf("Invalid choice! Defaulting to Small.\n");
        sizeChoice = 1;
    }

    // Sweet selection
    printf("\nAdd a Sweet:\n");
    for (int i = 0; i < 4; i++)
        printf("%d. %s (+Rs. %.2f)\n", i + 1, sweets[i], sweetPrices[i]);
    printf("Enter your choice (1-4): ");
    scanf("%d", &sweetChoice);

    if (sweetChoice < 1 || sweetChoice > 4)
    {
        printf("Invalid choice! No sweet added.\n");
        sweetChoice = 4;
    }

    // Calculate prices
    float drinkPrice = drinkPrices[drinkChoice - 1]; // -1 here because arrays start
with 0
    float sizePrice = sizePrices[sizeChoice - 1];
    float sweetPrice = sweetPrices[sweetChoice - 1];

    total = drinkPrice + sizePrice + sweetPrice + tempPrice;
    tax = total * TAX_RATE;
    finalTotal = total + tax;

    // Display summary
    printf("\n========================================\n");
    printf("          ORDER SUMMARY\n");
    printf("========================================\n");
    printf("Drink:          %s (Rs. %.2f)\n", drinks[drinkChoice - 1], drinkPrice);
    printf("Size:           %s (+Rs. %.2f)\n", sizes[sizeChoice - 1], sizePrice);
    printf("Sweet:          %s (+Rs. %.2f)\n", sweets[sweetChoice - 1], sweetPrice);
    printf("Temperature:    %s (+Rs. %.2f)\n", (tempChoice == 2) ? "Iced" : "Hot",
tempPrice);
    printf("-----------------------------------\n");
    printf("Subtotal:       Rs. %.2f\n", total); // Before tax
    printf("Tax (10%):      Rs. %.2f\n", tax);
    printf("-----------------------------------\n");
    printf("Total Bill:     Rs. %.2f\n", finalTotal);
    printf("=====================================\n");
    printf("Thank you! Your order has been received.\n");
    printf("Enjoy your drink!\n");
    printf("=====================================\n");

    return 0;
}
```

3.2 Code Explanation

The cafe billing system is written in the C programming language. It uses arrays, conditional statements, and basic arithmetic operations to generate a simple billing process for a cafe. The purpose of the code is to allow a user to choose a drink, select whether they want it hot or iced, decide the size, and add a sweetness level, after which the total bill is calculated with tax.

The program begins by displaying a **"***Welcome to Café***"** and then shows the menu of available drinks stored in an array. Each drink corresponds to a specific price. The user is asked to input their drink choice, and an if-else conditional checks whether the entered number is valid. If an invalid number is entered, the program automatically selects a default drink.

Next, the program prompts the user to select whether they want their drink Hot or Iced. This feature was added using another conditional statement. Each option slightly changes the price. For example, iced drinks come with an additional cost of Rs. 150 due to additional ingredients.

The user is then shown the size options i.e., small, medium, or large; each with its own price increment. Similarly, there's an option to add a sweet to your order. Invalid inputs are handled using default selections to prevent crashes.

Once all selections are made, the program calculates the total price by summing the base drink price, temperature addition, size cost, and sweetness charge. A sales tax of 10% is then applied, which is defined using a const variable at the start of the program. Using const ensures that the tax rate remains fixed and cannot be accidentally modified during execution of code.

After the calculations, the program displays a detailed bill summary showing:

1. The selected drink name
2. Temperature preference (Hot/Iced)
3. Size
4. Sweet (if added)
5. Subtotal before tax
6. Tax amount
7. Final total

This final output is neatly formatted using the printf() function to make it user-friendly and professional-looking. The program then displays a “Thank you! Your order has been received. Enjoy your drink!” message to signify the end of the transaction.

4. Future Enhancements

- Implement multiple order functionality to handle multiple customers simultaneously.
- Add **file handling** to save receipts and maintain order history.
- Include **discounts, loyalty points, and a membership system** for regular customers.
- Transform the console program into an interactive website using **HTML** for structure and **CSS** for styling.
- Introduce **error handling** and **input validation** for robust performance.
- Connect the system to a **database (MySQL or SQLite)** to store daily orders, total sales, and customer information.

5. Conclusion

The Cafe Billing System project successfully demonstrates the use of **arrays, switch statements, and arithmetic operations** to automate billing operations in a cafe environment.
It simplifies manual tasks, ensures accurate price computation, and provides a clear order summary to the user.
This project highlights the importance of **structured programming and logical**

flow control while reinforcing basic concepts of user interaction, decision making, and modular code design.

It serves as a foundational step toward developing more advanced and feature rich commercial applications in the future.