Software Tools And Technology

Group 7

:

Lab Notebook

Group members:

1. **Jaya Shree Biswas Bsc in IT(DS) (Leader)**
2. **Soumyadeep Goswami Bsc in IT(AI)**
3. **Suraj Maharaj BCA**
4. **Debapriya Dutta Bsc in IT(AI)**
5. **Koyena Brahma BCA**

Instructor: Dr.Ayan Ghosh
Course: Software Tools And Technology

Lab Notebook Entries

1 Lab Entry by Jaya Shree Biswas

1.1 Experiment

Sl. No.	Assignments
1.	Introduction to Github and Github desktop version installation

2 Lab Entry by Soumyadeep Goswami

2.1 Experiment

Sl. No.	Assignments
1.	Converting Submit button to Chin Tapak Dum Dum

3 Lab Entry by Suraj Maharaj

3.1 Experiment

Sl. No.	Assignments
1.	Making calculator in C

4 Lab Entry by Debapriya Dutta

4.1 Experiment

Sl. No.	Assignments
1.	Creating latex repository on github

5 Lab Entry by Koyena Brahma

5.1 Experiment

Sl. No.	Assignments
1.	Introduction to latex

Introduction to GitHub

GitHub is a web-based platform for version control using Git, enabling collaboration on software projects. It allows tracking changes, managing code, and working with others seamlessly. GitHub Desktop is a GUI tool that simplifies Git operations, making it easier for users to manage repositories without using the command line.

Installing GitHub Desktop

- **Download: Visit GitHub Desktop and download the version for your OS.**
- **Install: Run the installer and follow the prompts.**
- **Sign In: Open GitHub Desktop and sign in or create a GitHub account.**
- **Configure Git: Set your name and email for commits.**
- **Clone/Repository: Clone existing repositories or create a new one.**
- **Commit and Sync: Make changes, commit them, and push or pull updates from GitHub.**

GitHub Desktop streamlines Git operations, making version control accessible and straightforward.

Introduction to LaTeX

KOYENA BRAHMA

August 31, 2024

LaTeX

Contents

1 Introduction

LaTeX is a typesetting system that is widely used for producing scientific and mathematical documents due to its powerful handling of formulas and bibliographies. It is also used for other types of documents, from simple letters to complete books.

2 Basic Document Structure

A basic LaTeX document has the following structure:

```
\documentclass{article}
\begin{document}
% Your content here
\end{document}
```

3 Text Formatting

LaTeX provides various commands for text formatting. Here are some examples:

- **Bold Text** is created using \textbf{}.
- *Italic Text* is created using \textit{}.
- Underlined text can be created using \underline{}.

4 Mathematical Equations

One of the most powerful features of LaTeX is its ability to typeset complex mathematical equations. For example:

$$E = mc^2 \tag{1}$$

Inline equations can be written using the $ symbol, like this: $a^2 + b^2 = c^2$.

5 Inserting Images

You can include images in your LaTeX document using the graphicx package. Here's an example:

```
\begin{figure}[h]
    \centering
    \includegraphics[width=0.5\textwidth]{example-image}
    \caption{An example image.}
    \label{fig:example}
\end{figure}
```

6 Creating Lists

LaTeX allows you to create both numbered and bulleted lists easily.

6.1 Bulleted List

- First item
- Second item
- Third item

6.2 Numbered List

1. First item
2. Second item
3. Third item

7 Adding Hyperlinks

You can add hyperlinks in your document using the `hyperref` package. For example:

Visit the LaTeX project website.

8 Conclusion

This document provides a brief introduction to some of the basic features of LaTeX. There are many more advanced features that can help you create professional-looking documents.

Changing the submit button to Chin Tapak Dum Dum and fixing the disproportionate

1 STEPS

Changed the submit button to"Chin tapak Dum Dum through this code.

```
add(symbolPanel, BorderLayout.CENTER);

// Panel for submit button
Panel controlPanel = new Panel(new FlowLayout());
submitButton = new Button("Chin Tapak Dum Dum");
submitButton.setFont(new Font("Arial", Font.BOLD, 20));
submitButton.setBackground(Color.RED);
submitButton.setForeground(Color.WHITE);
submitButton.addActionListener(this);
controlPanel.add(submitButton);
add(controlPanel, BorderLayout.SOUTH);
```

Figure 1: JAVA CODE

- **Font Size and Style:** The font size and style have been adjusted for improved readability and consistency with the overall design.
- **Background Color:** The background color of the button has been updated to create a more visually appealing and cohesive look.
- **Font Color:** The font color has been modified to ensure strong contrast with the background, enhancing legibility.
- **Element Proportions:** Any disproportionate elements have been corrected to achieve a more balanced and aesthetically pleasing design.

2 OUTPUT

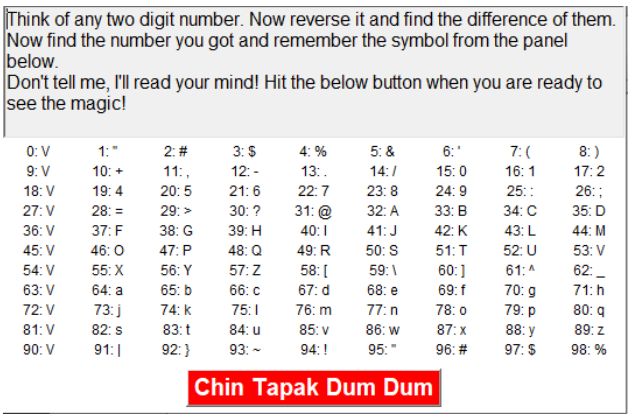

Figure 2: FINAL OUTPUT

Calculator in C

Suraj Maharaj

September 9, 2024

1 Introduction

In this document, I will present a detailed explanation of a simple calculator program developed in the C programming language. This calculator performs various arithmetic operations including addition, subtraction, multiplication, division, percentage calculation, squaring, and cubing of numbers. The design of this calculator is aimed at providing a user-friendly interface with robust input validation to ensure accurate results.

2 Code

Below is the complete code for the calculator in C , followed by an explanation of each function and its role in the program.

Listing 1: Simple Calculator in C

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

// Function declarations
void addition();
void subtract();
void multiply();
void divide();
void percentage();
void square();
void cube();

int main() {
    int op;

    do {
        printf("Select an operation to perform in the C Calculator
            :\n");
        printf("1. Addition\n");
        printf("2. Subtraction\n");
        printf("3. Multiplication\n");
        printf("4. Division\n");
        printf("5. Percentage\n");
        printf("6. Square\n");
        printf("7. Cube\n");
        printf("8. Exit\n");
```

```
        printf("Please make a choice: ");

        // Validate user input
        while (scanf("%d", &op) != 1) {
            printf("Invalid input! Please enter a number between 1
                and 8: ");
            while (getchar() != '\n'); // Clear the invalid input
        }

        // Perform the selected operation
        switch (op) {
            case 1:
                addition(); // Call the addition function
                break;
            case 2:
                subtract(); // Call the subtraction function
                break;
            case 3:
                multiply(); // Call the multiplication function
                break;
            case 4:
                divide(); // Call the division function
                break;
            case 5:
                percentage(); // Call the percentage function
                break;
            case 6:
                square(); // Call the square function
                break;
            case 7:
                cube(); // Call the cube function
                break;
            case 8:
                printf("Exiting the program.\n");
                exit(0); // Exit the program
            default:
                printf("Error! Invalid choice. Try again.\n");
        }
        printf("\n*****************************************\n");
    } while (op != 8); // Repeat until the user chooses to exit

    return 0;
}

// Function definitions

// Function to add numbers
void addition() {
    int i, num;
    double sum = 0;
    printf("How many numbers do you want to add? ");

    // Read the number of inputs and validate it
    while (scanf("%d", &num) != 1 || num <= 0) {
        printf("Invalid input! Enter a positive number: ");
        while (getchar() != '\n'); // Clear the invalid input
    }

```

```
    printf("Enter the numbers:\n");
    for (i = 1; i <= num; i++) {
        double f_num;
        while (scanf("%lf", &f_num) != 1) {
            printf("Invalid input! Enter a valid number: ");
            while (getchar() != '\n'); // Clear the invalid input
        }
        sum += f_num;
    }

    // Display the result
    printf("Total sum of the numbers = %.2lf\n", sum);
}

// Function to subtract two numbers
void subtract() {
    double n1, n2;
    printf("Enter the first number: ");
    scanf("%lf", &n1);
    printf("Enter the second number: ");
    scanf("%lf", &n2);
    printf("The result of %.2lf - %.2lf is: %.2lf\n", n1, n2, n1 -
        n2);
}

// Function to multiply two numbers
void multiply() {
    double n1, n2;
    printf("Enter the first number: ");
    scanf("%lf", &n1);
    printf("Enter the second number: ");
    scanf("%lf", &n2);
    printf("The result of %.2lf * %.2lf is: %.2lf\n", n1, n2, n1 *
        n2);
}

// Function to divide two numbers
void divide() {
    double n1, n2;
    printf("Enter the first number: ");
    scanf("%lf", &n1);
    printf("Enter the second number: ");
    scanf("%lf", &n2);

    if (n2 != 0) {
        printf("The result of %.2lf / %.2lf is: %.2lf\n", n1, n2,
            n1 / n2);
    } else {
        printf("Error! Division by zero is not allowed.\n");
    }
}

// Function to calculate the percentage
void percentage() {
    double value, percent;
    printf("Enter the value: ");
    scanf("%lf", &value);
    printf("Enter the percentage: ");
```

```
139     scanf("%lf", &percent);
140     printf("%.2lf percent of %.2lf is: %.2lf\n", percent, value, (
            percent / 100) * value);
141 }
142
143 // Function to calculate the square of a number
144 void square() {
145     double n1;
146     printf("Enter a number to get the square: ");
147     scanf("%lf", &n1);
148     printf("The square of %.2lf is: %.2lf\n", n1, n1 * n1);
149 }
150
151 // Function to calculate the cube of a number
152 void cube() {
153     double n1;
154     printf("Enter a number to get the cube: ");
155     scanf("%lf", &n1);
156     printf("The cube of %.2lf is: %.2lf\n", n1, n1 * n1 * n1);
157 }
```

3 Function Explanations With Output Values

3.1 Addition Function

The `addition()` function allows the user to input multiple numbers and calculates their sum. It prompts the user for the number of inputs, validates the input, and then reads the numbers, summing them up.

Example Output:

```
Select an operation to perform in the C Calculator:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Square
7. Cube
8. Exit
Please make a choice: 1
How many numbers do you want to add? 3
Enter the numbers:
5
10
15
Total sum of the numbers = 30.00
```

3.2 Subtraction Function

The `subtract()` function performs subtraction between two user-provided numbers. It prompts the user for two numbers and then calculates and displays their difference.

Example Output:

```
Select an operation to perform in the C Calculator:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Square
7. Cube
8. Exit
Please make a choice: 2
Enter the first number: 20
Enter the second number: 5
The result of 20.00 - 5.00 is: 15.00
```

3.3 Multiplication Function

The `multiply()` function performs multiplication of two numbers entered by the user. It displays the product of the two numbers.

Example Output:

```
Select an operation to perform in the C Calculator:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Square
7. Cube
8. Exit
Please make a choice: 3
Enter the first number: 4
Enter the second number: 5
The result of 4.00 * 5.00 is: 20.00
```

3.4 Division Function

The `divide()` function handles the division of two numbers. It includes a check to prevent division by zero, ensuring the user does not encounter an error.

Example Output:

```
Select an operation to perform in the C Calculator:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Square
7. Cube
8. Exit
Please make a choice: 4
Enter the first number: 25
Enter the second number: 5
The result of 25.00 / 5.00 is: 5.00
```

3.5 Percentage Function

The `percentage()` function calculates the percentage of a given value. The user inputs a value and the percentage, and the function computes and displays the result.

Example Output:

```
Select an operation to perform in the C Calculator:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Square
7. Cube
8. Exit
Please make a choice: 5
Enter the value: 200
Enter the percentage: 15
15.00 percent of 200.00 is: 30.00
```

3.6 Square Function

The `square()` function calculates the square of a number. The user inputs a number, and the function computes and displays its square.

Example Output:

```
Select an operation to perform in the C Calculator:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Square
```

```
7. Cube
8. Exit
Please make a choice: 6
Enter a number to get the square: 7
The square of 7.00 is: 49.00
```

3.7 Cube Function

The `cube()` function calculates the cube of a number. The user inputs a number, and the function computes and displays its cube.

Example Output:

```
Select an operation to perform in the C Calculator:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Square
7. Cube
8. Exit
Please make a choice: 7
Enter a number to get the cube: 3
The cube of 3.00 is: 27.00
```

4 Conclusion

In conclusion, the simple calculator program developed in C provides a comprehensive tool for performing basic arithmetic operations. This calculator covers a range of functions including addition, subtraction, multiplication, division, percentage calculation, and the computation of squares and cubes of numbers. Each function is designed with user-friendly prompts and robust input validation to ensure accuracy and ease of use.

The program demonstrates effective use of functions to modulates the code, making it both organized and easy to maintain. By implementing input validation and error handling, the calculator minimizes the risk of user errors and enhances the overall user experience.

This project not only highlights fundamental programming concepts such as function definition and control structures but also showcases practical application in developing tools for everyday use. This simple calculator serves as a solid foundation for building more complex applications and improving programming skills in C.

Overall, this calculator is a valuable exercise in programming, offering insights into both basic and intermediate concepts, and providing a useful utility for performing arithmetic operations efficiently.

.