MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

MAKAUT WB

Software Tools and Technology

Lab Notebook

submitted by: - **Group 25**
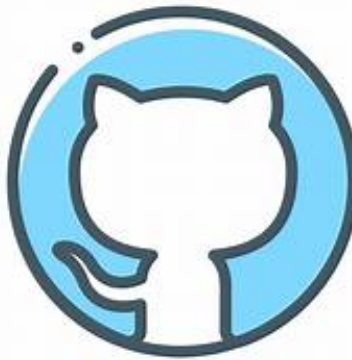
Group Members

1. **Subhankar Prasad Harijan** (**LEADER**)
   Department: - BCA
   Roll No: - 30001223005

2. **Gloria Ghosh**
   Department: - BSc in IT (Data Science)
   Roll No: - 30084323024

3. **Sresta Tewari**
   Department: - BSc in IT (Artificial Intelligence)
   Roll No: - 30054623004

4. **Sourav Sasmal**
   Department: - BCA
   Roll No: - 30001223073

5. **Aakash Roy**
   Department: - BSc in IT (Data Science)
   Roll No: - 30084323015

INDEX

Lab Assignment-1

Introduction to GitHub and GitHub Desktop version installation

GitHub

GitHub is a web-based platform that allows developers to host, share, and collaborate on software projects. It provides a version control system powered by Git, enabling teams to track changes, manage code repositories, and work together seamlessly across different locations. GitHub supports collaborative development through features like pull requests, issues, and project boards, making it essential for open-source projects and professional software development. Additionally, it integrates with various development tools, enhancing productivity and streamlining the software development lifecycle.

Installation

Installing GitHub Desktop is a straightforward process that enhances your workflow by providing a user-friendly interface for managing repositories. To begin, download the installer from the official GitHub Desktop website for your operating system—Windows or macOS. After downloading, run the installer and follow the on-screen instructions to complete the setup. Once installed, launch the application and sign in with your GitHub credentials, or create a new account if needed. GitHub Desktop simplifies the process of cloning repositories, making commits, and managing branches, making it an invaluable tool for developers of all skill levels. For Linux users, alternative methods like using Wine or other Git clients are available.

<div align="center">

Lab Assignment-2

</div>

1 Calculator Program using C

1.1 Objective

The objective of this lab is to develop a basic calculator program using the C programming language. The calculator will perform simple arithmetic operations like addition, subtraction, multiplication, and division based on user input.

1.2 Program Overview

The calculator program is designed to:

- Accept two numbers from the user.

- Prompt the user to select an arithmetic operation (Addition, Subtraction, Multiplication, Division).

- Perform the selected operation.

- Display the result of the operation to the user.

The program includes error handling to manage division by zero and other invalid inputs.

1.3 Code Implementation

The following is the C code for the calculator program:

```c
#include<stdio.h>
#include<conio.h>
void main(){
    float a,b,c;
    char ch;
    printf("Enter the first number : ");
    scanf("%f",&a);
    printf("Enter user choice operation : ");
    scanf(" %c",&ch);
    printf("Enter the second number : ");
    scanf("%f",&b);
    switch(ch){
        case'+':c=a+b;
        printf("Output is : %f",c);
        break;
        case'-':c=a-b;
        printf("Output is : %f",c);
        break;
        case'*':c=a*b;
        printf("Output is : %f",c);
        break;
        case'/':c=a/b;
        printf("Output is : %f",c);
        break;
        default:printf("Invalid operation");
        break;
    }
    getch();
```

```
}
```

1.4 Compiling and Running the Program

To compile and run the calculator program:

1. Open a terminal or command prompt.

2. Navigate to the directory where the C file is located.

3. Compile the program using a C compiler (e.g., GCC):

```
gcc calculator.c -o calculator
```

4. Run the compiled program:

```
./calculator
```

1.5 Adding the Calculator Program to GitHub Repository

To add this calculator program to a GitHub repository, follow these
steps:

1.5.1 Step 1: Initialize a Local Git Repository

1. Open the terminal and navigate to the directory where your
   `calculator.c` file is located.

2. If you haven't already, initialize a Git repository in that directory:

```
git init
```

This command creates a new Git repository in the current direc-
tory.

1.5.2   Step 2: Add the File to the Repository

1. Add the `calculator.c` file to the staging area:

   ```
   git add calculator.c
   ```

   This command stages the file, indicating that you want to include it in the next commit.

1.5.3   Step 3: Commit the Changes

1. Commit the file to the repository with a meaningful message:

   ```
   git commit -m "Add calculator program in C"
   ```

1.5.4   Step 4: Push the Changes to GitHub

1. Link your local repository to a remote GitHub repository:

   ```
   git remote add origin https://github.com/yourusername/your-repo-name.git
   ```

2. Push the changes to the GitHub repository:

   ```
   git push -u origin master
   ```

1.5.5   Step 5: Verify the Upload

1. Go to your GitHub repository URL in a web browser.

2. Verify that the `calculator.c` file is listed and accessible in the repository.

<center>**Lab Assignment-3**</center>

<center>**How to create Matrix in LaTeX**</center>

To create a matrix in LaTeX, we use the `amsmath` package, which provides various environments for displaying matrices. Here's a basic guide on how to create a matrix and the different options available:

- **Include the `amsmath` Package**

First, ensure you have the `amsmath` package included in your LaTeX document preamble. Add the following line:

`\usepackage{amsmath}`

- **Matrix Environments**

There are several environments for creating matrices, depending on how you want them formatted. Here are the most common ones:

- `matrix`: A basic matrix without brackets.

- `bmatrix`: A matrix with square brackets.

- `pmatrix`: A matrix with parentheses.

- `vmatrix`: A matrix with vertical bars.

- `Vmatrix`: A matrix with double vertical bars.

<center>**Syntax**</center>

The syntax for creating a matrix is similar across different environments. Use the `\begin{environment}` and `\end{environment}`

<center>8</center>

commands to enclose the matrix content. Separate the elements in each row with & and
 end each row with
.

Here's an example of a 2x2 matrix in each environment:

```
[
$ \begin{matrix}
    1 & 0 \\
    0 & 1
\end{matrix}
$ ]


\begin{bmatrix}    [
 $   \begin{matrix}
    1 & 0\\
    0 & 1
    \end{matrix}
 $   ]


a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}


\begin{pmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{pmatrix}


\begin{vmatrix}
```

```
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{vmatrix}

\begin{Vmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{Vmatrix}
```

Explanation

- `\begin{matrix} ... \end{matrix}`: Creates a matrix with no surrounding brackets.

- `\begin{bmatrix} ... \end{bmatrix}`: Surrounds the matrix with square brackets.

- `\begin{pmatrix} ... \end{pmatrix}`: Surrounds the matrix with parentheses.

- `\begin{vmatrix} ... \end{vmatrix}`: Surrounds the matrix with single vertical bars, often used to denote determinants.

- `\begin{Vmatrix} ... \end{Vmatrix}`: Surrounds the matrix with double vertical bars, also used for determinants or norms.

We can adjust the number of rows and columns by adding or removing elements and rows as needed.

<u>**Lab Assignment-4**</u>

<u>**How to create section, subsection, and subsubsection in LaTeX.**</u>

<u>Introduction</u>

In LaTeX, sections, subsections, and subsubsections are used to organize and structure your document. Here's a brief guide on how to create and format these hierarchical levels:

2   <u>Basic Structure</u>

2.1   Section

- Created using the \section{} command.
- The section title is automatically numbered.

2.2   Subsection

- Created using the \subsection{} command.
- Falls under the section and is automatically numbered as a sub-level.

2.3   Subsubsection

- Created using the \subsubsection{} command.
- Falls under the subsection and is automatically numbered as a sub-sub-level.

3 Example

Here's a simple example to demonstrate the usage of sections, subsections, and subsubsections:

```
\documentclass{article}

\begin{document}

\title{Document with Sections}
\author{Author Name}
\date{\today}
\maketitle

\section{Introduction}
This is the introduction section. Here we discuss the background and context of the document.

\subsection{Motivation}
This subsection explains the motivation behind the topic. It provides insights into why the topic is relevant.

\subsubsection{Historical Context}
This subsubsection delves into the historical background of the topic, providing a timeline of important events.

\subsection{Objectives}
This subsection outlines the objectives and goals of the document.

\section{Methods}
This section describes the methods used in the study or analysis.

\subsection{Data Collection}
Details about how the data was collected are discussed here.

\subsubsection{Survey Method}
This subsubsection provides specifics on the survey method used for data collection.

\section{Results}
This section presents the results of the study or analysis.

\subsection{Analysis}
Details about the analysis of the results are provided in this subsection.

\subsubsection{Statistical Analysis}
This subsubsection focuses on the statistical methods used to analyze the data.

\section{Conclusion}
The final section summarizes the findings and provides conclusions.

\end{document}
```

4 Explanation

- \section{}: Creates a new section and numbers it accordingly. It is the highest level of sectioning in the article class.
- \subsection{}: Creates a subsection under the current section. It is numbered as part of the section number (e.g., 1.1, 1.2).
- \subsubsection{}: Creates a subsubsection under the current subsection. It is numbered as part of the subsection number (e.g., 1.1.1, 1.1.2).

<div align="center">

Lab Assignment-5

Enumerate ABC Format and Roman Number in LaTeX

Introduction

</div>

To create enumerated lists in LaTeX using both alphabetic (ABC) format and Roman numeral format, you can use the `enumerate` environment. LaTeX allows customization of list labels either by using the `enumerate` package (basic customization) or the `enumitem` package (more advanced customization). Below is a detailed explanation and corresponding LaTeX code.

1.Alphabetic Format (ABC):

- For alphabetic enumeration (A, B, C...), you can use either the `enumerate` package or the `enumitem` package.

2. Roman Numerals:

- For Roman numeral enumeration (I, II, III...), you can use `[I.]` with the `enumerate` package, or `label=.` with the `enumitem` package.

Example

Here's the LaTeX code that implements both alphabetic and Roman numeral enumerations:

```
\documentclass{article}

% Using the enumerate package for basic list formatting
\usepackage{enumerate}

% Using the enumitem package for advanced list formatting
\usepackage{enumitem}

\begin{document}

\section*{Enumerated Lists: ABC Format and Roman Numerals}

\subsection*{Using the enumerate Package}

\subsubsection*{Alphabetic Format (ABC)}

% Alphabetic enumeration using enumerate package
\begin{enumerate}[A.]
  \item First item
  \item Second item
```

```latex
    \item Third item
\end{enumerate}

\subsubsection*{Roman Numerals}

% Roman numeral enumeration using enumerate package
\begin{enumerate}[I.]
  \item First item
  \item Second item
  \item Third item
\end{enumerate}

\subsection*{Using the enumitem Package}

\subsubsection*{Alphabetic Format (ABC)}

% Alphabetic enumeration using enumitem package
\begin{enumerate}[label=\Alph*.]
  \item First item
  \item Second item
  \item Third item
\end{enumerate}

\subsubsection*{Roman Numerals}

% Roman numeral enumeration using enumitem package
\begin{enumerate}[label=\Roman*.]
  \item First item
  \item Second item
  \item Third item
\end{enumerate}

\end{document}
```

Explanation

1. Alphabetic Format (ABC):

- **Using the enumerate package:** In the `enumerate` environment, you can specify the list type as `[A.]` to generate an alphabetic list (A, B, C...).
- **Using the enumitem package:** The `enumitem` package allows more flexibility by setting the label format as `label=.`.

2. **Roman Numerals:**

- **Using the enumerate package:** For Roman numerals, you use `[I.]` to create the list items as I, II, III...

- **Using the enumitem package:** You can customize the label with `label=.` to achieve the same effect with more control over spacing and style.

Output:

The compiled LaTeX document will display two types of lists (ABC and Roman numerals) using both `enumerate` and `enumitem` packages.