

Practical 2

Aim: Execute C Program using gcc compiler.

Step 1. Open up a terminal.

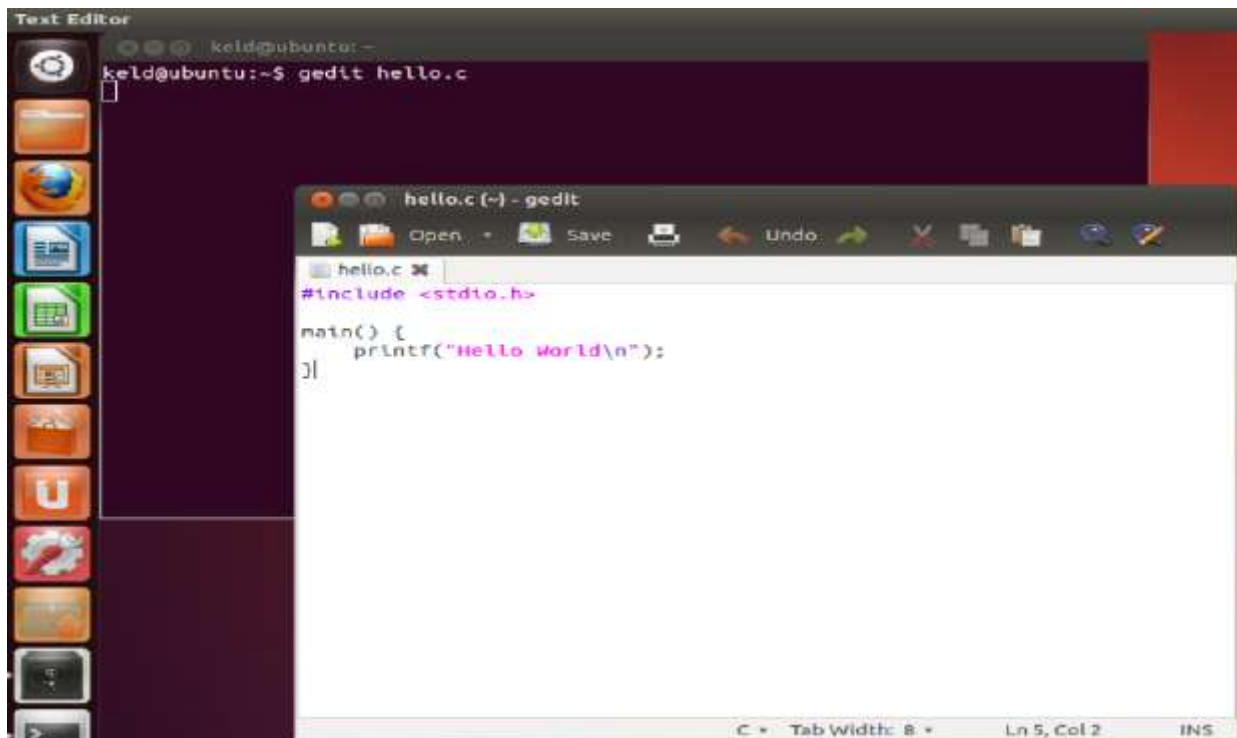


Step 2. Use a text editor to create the C source code.

Type the command `$gedit hello.c` And enter the C source code shown below.

```
#include<stdio.h>

int main()
{
    printf("Hello World");
    return 0;
}
```



The screenshot shows a Linux desktop environment. In the background, a terminal window titled 'keld@ubuntu:~\$' displays the command 'gedit hello.c'. In the foreground, a text editor window titled 'hello.c (~) - gedit' is open, showing the following C code:

```
#include <stdio.h>

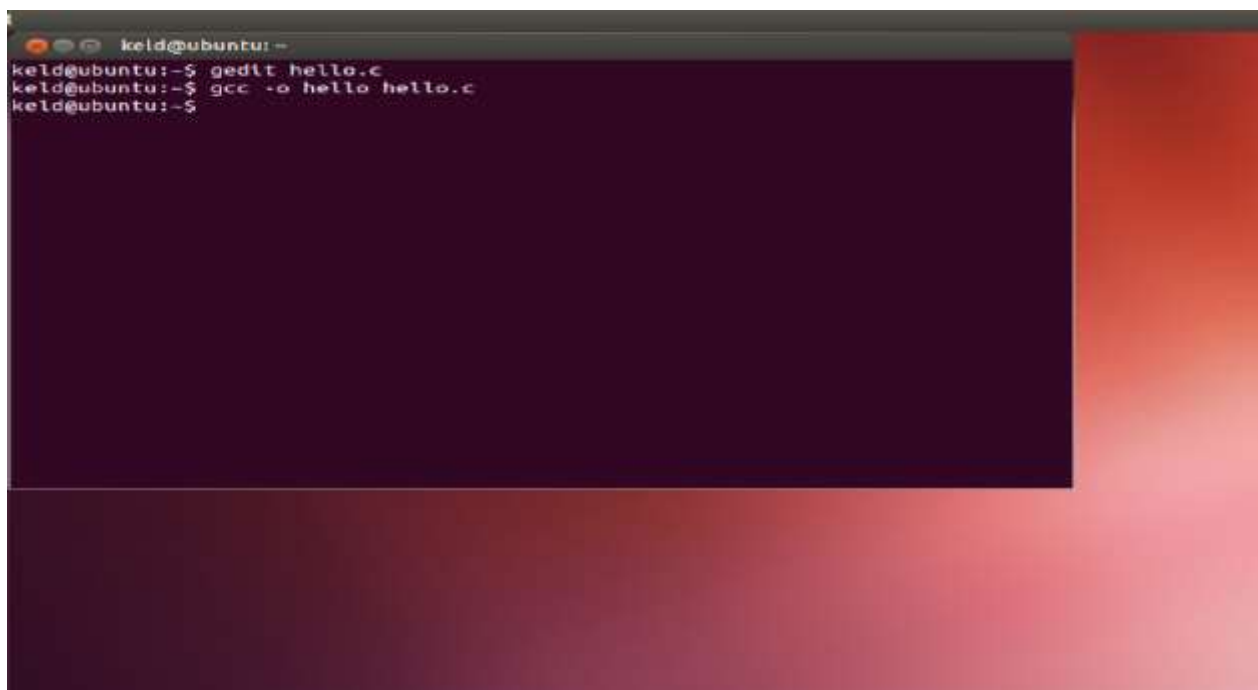
main() {
    printf("Hello World\n");
}
```

The text editor window has a menu bar with 'Open', 'Save', 'Undo', and other options. The status bar at the bottom indicates 'C * Tab Width: 8 * Ln 5, Col 2 INS'.

Step 3: Compile the program.

Type the command **\$gcc hello.c**

This command will invoke the GNU C compiler to compile the file `hello.c` and result in an executable called `hello`.



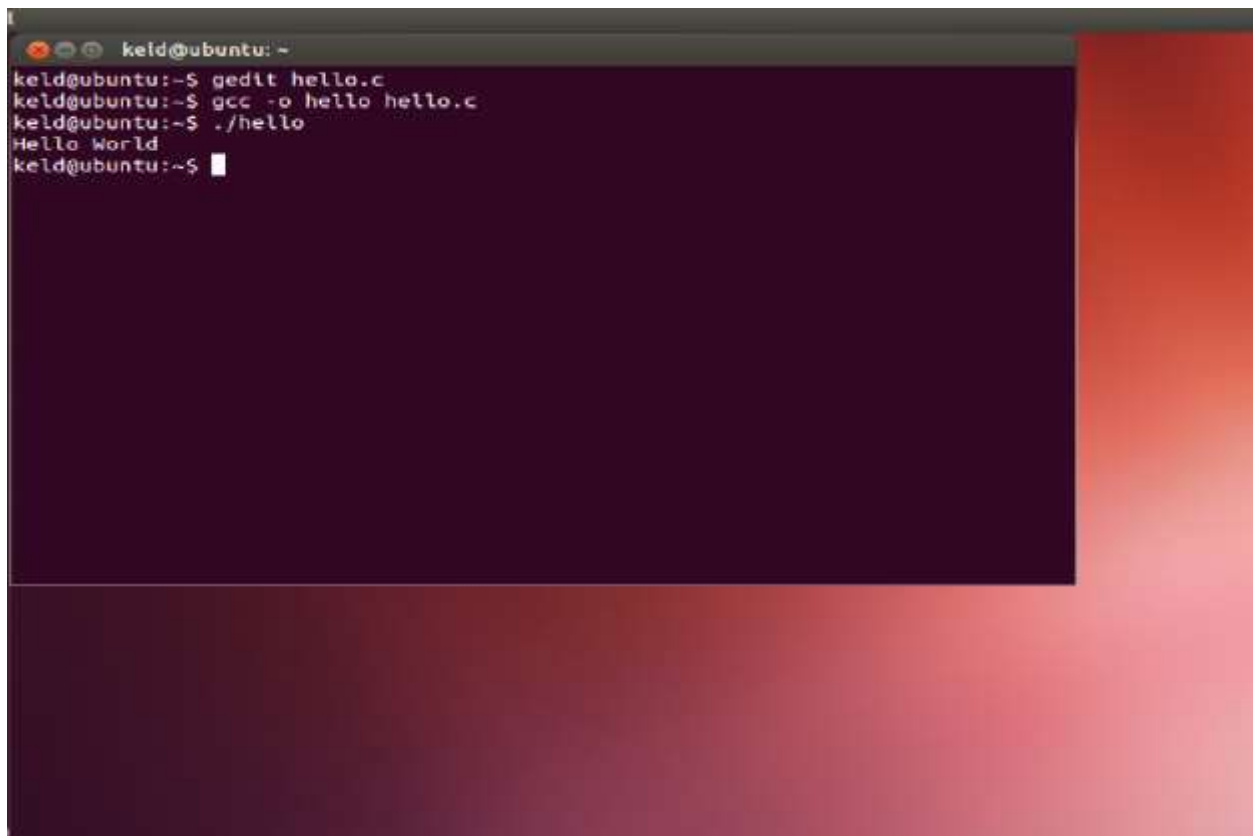
The screenshot shows a terminal window titled 'keld@ubuntu:~\$'. The following commands and their outputs are shown:

```
keld@ubuntu:~$ gedit hello.c
keld@ubuntu:~$ gcc -o hello hello.c
keld@ubuntu:~$
```

Step 4: Execute the program.

Type the command **./a.out**

This would result in the output: Hello World



```
keld@ubuntu: ~  
keld@ubuntu:~$ gedit hello.c  
keld@ubuntu:~$ gcc -o hello hello.c  
keld@ubuntu:~$ ./hello  
Hello World  
keld@ubuntu:~$
```

Conclusion:

In conclusion, executing a C program using the gcc compiler involves writing the code, compiling it with gcc, and running the resulting executable. This straightforward process ensures efficient development and testing of C programs on various platforms.

Practical 3

AIM: Demonstration of gpr of command using Linux.

THEORY:

The objective of profiling is to analyze your program code and see which part of the code is taking a large amount of time for execution such that the part of the code can be rewritten. This will enable the program to achieve desired execution speed. In addition, profiling can prove to be very handy in spotting codes that are potentially error-prone, and then they can be sorted out via refactoring.

Using the gprof is quite simple. First, you need to enable profiling when you compile the code. Now, when you execute the program, profile data is generated. Finally, you can run the gprof tool on the profiling data generated during execution. This will produce an analysis file. This file contains several tables, including a flat profile and a call graph.

Enable profiling while compiling

To enable profiling while compiling, simply add the -pg option in the compilation step. The pg flag generates extra code for profiling that is used by the gprof command. The following command illustrates how we can compile profiling information:

```
$ gcc -Wall -pg program.c program_new.c -o program
```

Execute the code to generate profile information

The binary file generated above can be executed to generate profile information. The following line executes the code:

```
./program
```

This will also generate a file gmon.out in the current working directory. To see what files are generated via execution of the above command, simply type ls as follows:

```
$ls
```

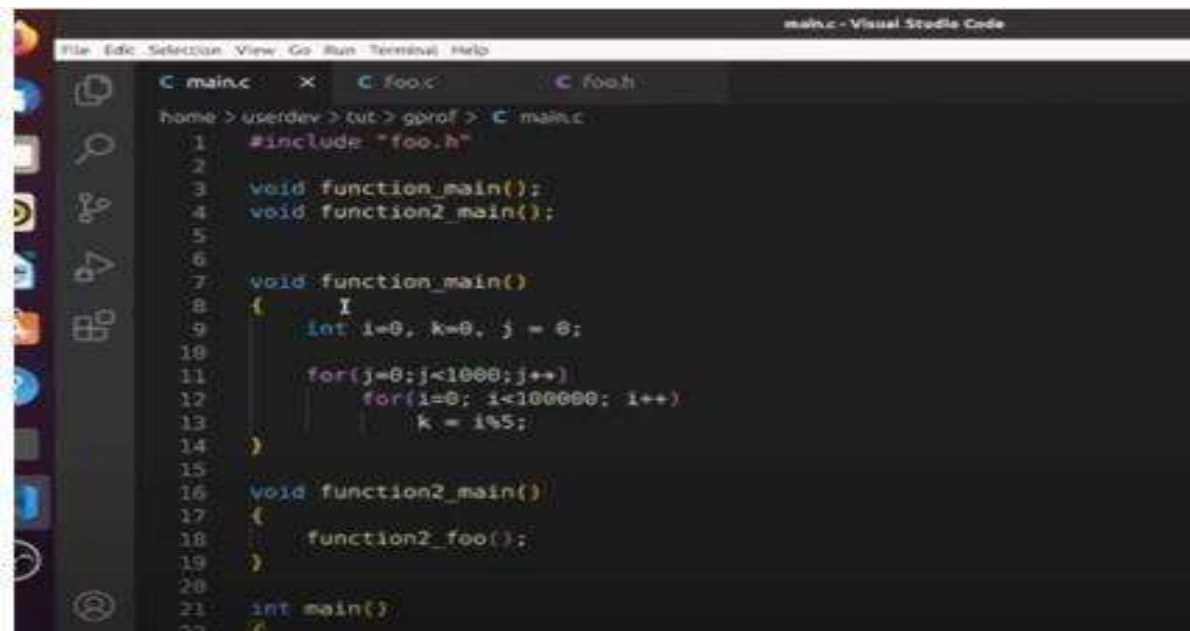
Run the gprof tool

Now, we will run the gprof tool providing as an argument the output file and gmon.out file. This will produce the profiling information:

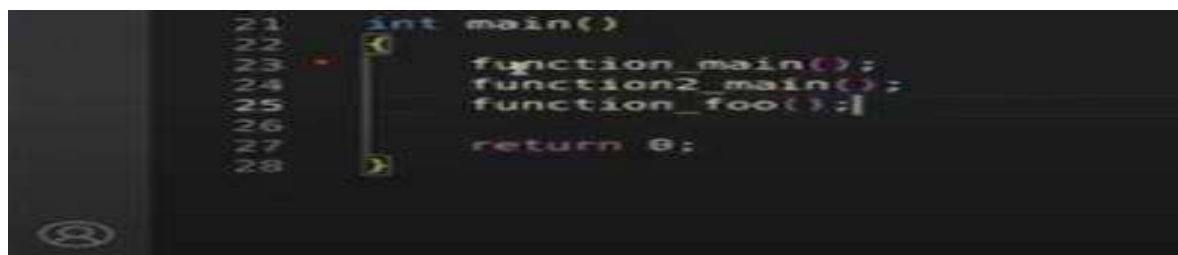
```
$ gprof program gmon.out > analysis.txt
```

PROGRAM CODE:

main.c file

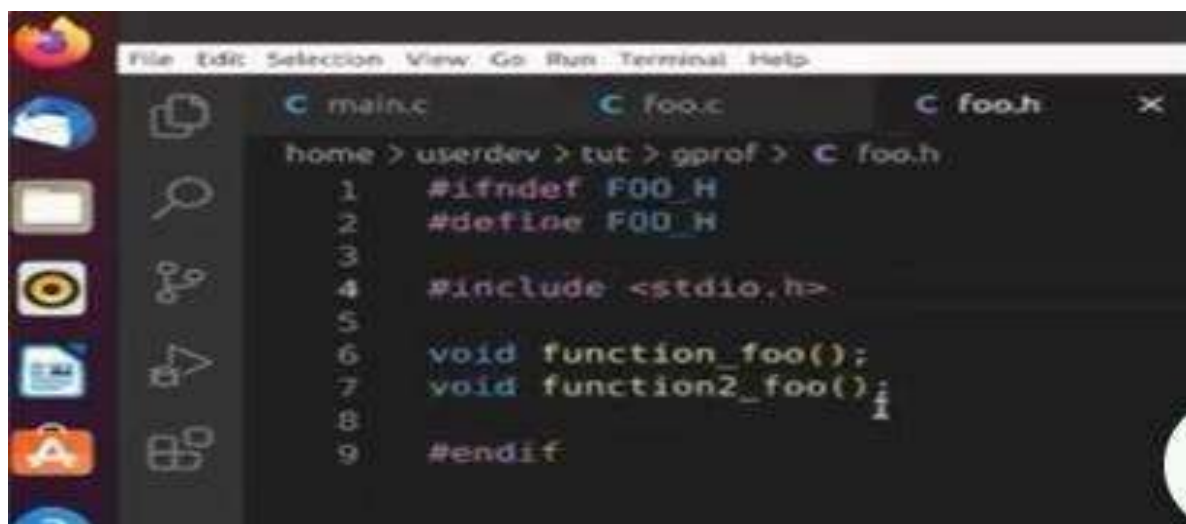


```
home > userdev > tut > gprof > C main.c
1  #include "foo.h"
2
3  void function_main();
4  void function2_main();
5
6
7  void function_main()
8  {
9      I
10     int i=0, k=0, j = 0;
11
12     for(j=0;j<1000;j++)
13         for(i=0; i<100000; i++)
14             k = i%5;
15
16 void function2_main()
17 {
18     function2_foo();
19 }
20
21 int main()
22 {
```



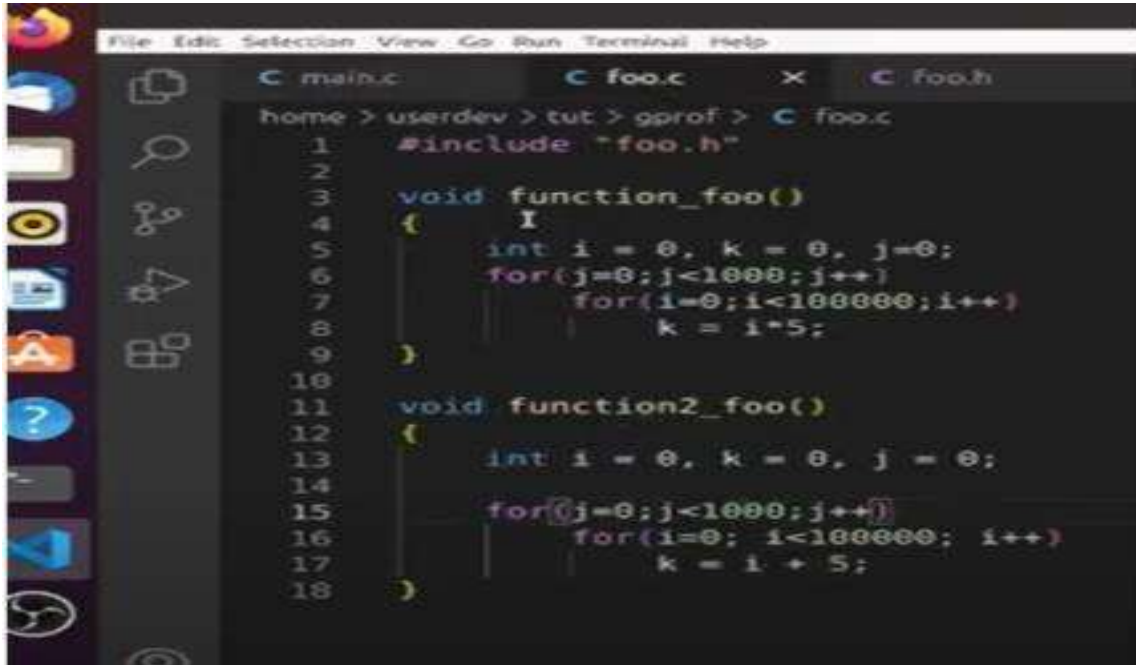
```
21 int main()
22 {
23     function_main();
24     function2_main();
25     function_foo();
26
27     return 0;
28 }
```

foo.c file:



```
home > userdev > tut > gprof > C foo.c
1  #ifndef FOO_H
2  #define FOO_H
3
4  #include <stdio.h>
5
6  void function_foo();
7  void function2_foo();
8
9  #endif
```

Foo.h file



```

File Edit Selection View Go Run Terminal Help
C main.c C foo.c X C foo.h
home > userdev > tut > gprof > C foo.c
1  #include "foo.h"
2
3  void function_foo()
4  {
5      int i = 0, k = 0, j=0;
6      for(j=0;j<1000;j++)
7          for(i=0;i<100000;i++)
8              k = i*5;
9  }
10
11 void function2_foo()
12 {
13     int i = 0, k = 0, j = 0;
14
15     for(j=0;j<1000;j++)
16         for(i=0; i<100000; i++)
17             k = i + 5;
18 }
  
```

OUTPUT:



```

Picture in picture
children  This is the amount of time that was propagated from the
          child's children to the function.

called    This is the number of times the function called
          this child '/' the total number of times the child
          was called. Recursive calls by the child are not
          listed in the number after the '/'.

name      This is the name of the child. The child's index
          number is printed after it. If the child is a
          member of a cycle, the cycle number is printed
          between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The '+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
  
```



```

userdev@device: ~/tut/gprof
[...]
```

```

[...no-]flat-profile[=name]] [...no-]graph[=name]]
[...no-]time[=name]] [...all-lines]] [...brief]] [...debug[=level]]
[...function-ordering]] [...file-ordering]] [...inline-file-names]]
[...directory-path=wdir]] [...display-unused-functions]]
[...file-format=name]] [...file-info]] [...help]] [...line]] [...min-count=n]]
[...no-static]] [...print-path]] [...separate-files]]
[...static-call-graph]] [...sun]] [...table-length=len]] [...traditional]]
[...version]] [...width=n]] [...ignore-non-functions]]
[...denangle=STYLE]] [...no-denangle]] [...external-symbol-table=name]] [%FILE]
[info-file]] [profile-file...]

```

Report bugs to <<http://www.sourceware.org/bugzilla/>>

```

userdev@device: ~/tut/gprof$ gprof gnmon.out main > report.txt
gprof: gnmon.out: not in executable format
userdev@device: ~/tut/gprof$ gprof main gnmon.out > report.txt
userdev@device: ~/tut/gprof$ ls
foo.c foo.h gnmon.out main main.c report.txt
userdev@device: ~/tut/gprof$ cat report.txt
Flat profile:

Each sample counts as 0.01 seconds.

```

%	cumulative	self		self	total	
time	seconds	seconds	calls	ns/call	ns/call	name
45.60	0.09	0.09	1	91.20	91.20	function_main
30.40	0.15	0.06	1	60.80	60.80	function_foo
25.33	0.20	0.05	1	50.67	50.67	function2_foo
0.00	0.20	0.00	1	0.00	50.67	function2_main

% the percentage of the total running time of the program used by this function.

cumulative a running sum of the number of seconds accounted for by this function and those listed above it.

self the number of seconds accounted for by this function alone. This is the major sort for this listing.



Faculty of Engineering & Technology

Subject: Open Source Software LAB

Subject-Code: 303105103

B-TECH: IT Year: 1st Sem: 1st

Conclusion:

In conclusion, the gpr command in Linux is a utility for managing print jobs and sending files to printers. Demonstrating its usage helps users understand how to print files efficiently, specify printer options, and manage the printing process directly from the terminal.

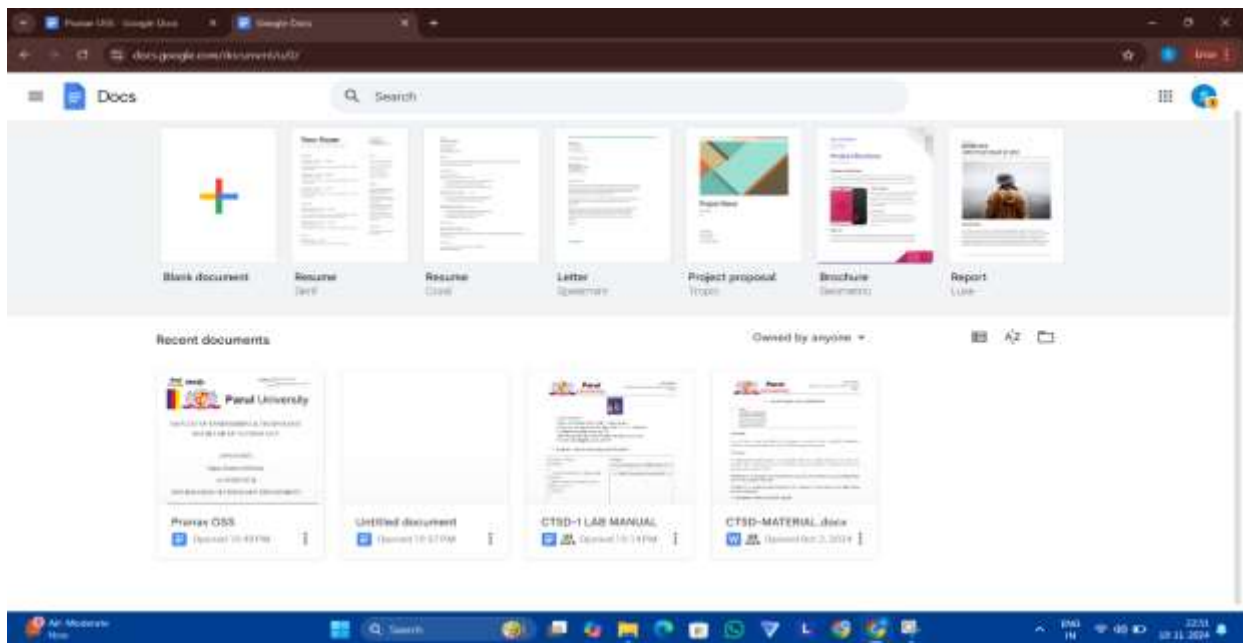
Practical 4

AIM: Create and Edit documents using Google Docs.

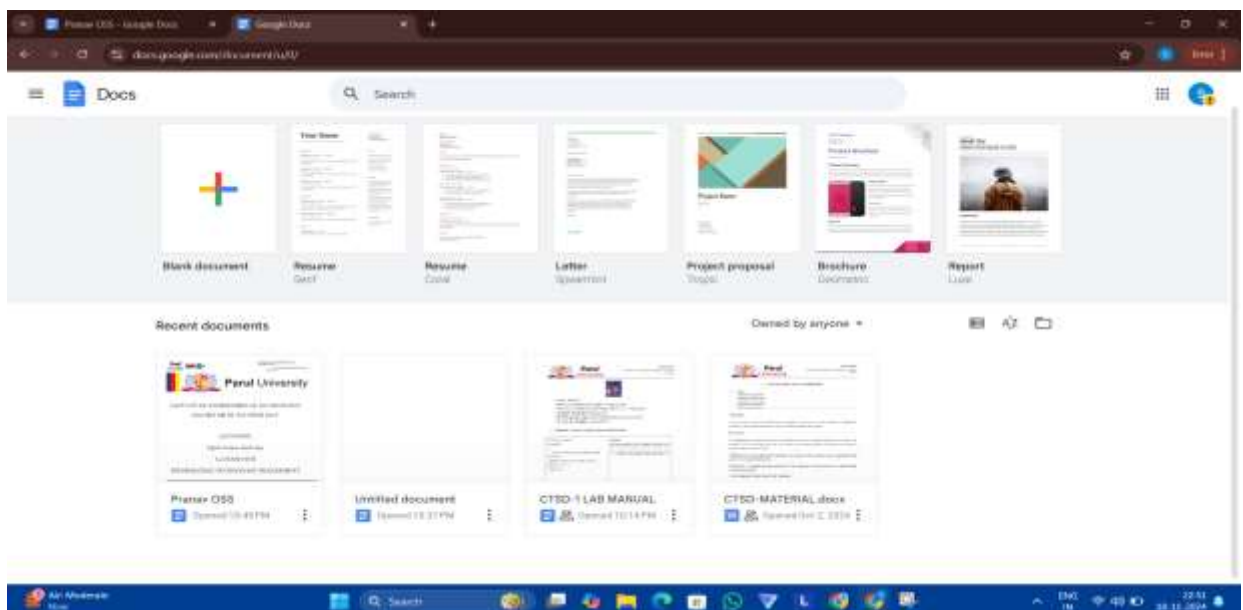
Step 1: Create a document

To create a new document:

1. On your computer, open the Docs home screen at docs.google.com.



2. In the top left, under "start a new document", click Blank New.



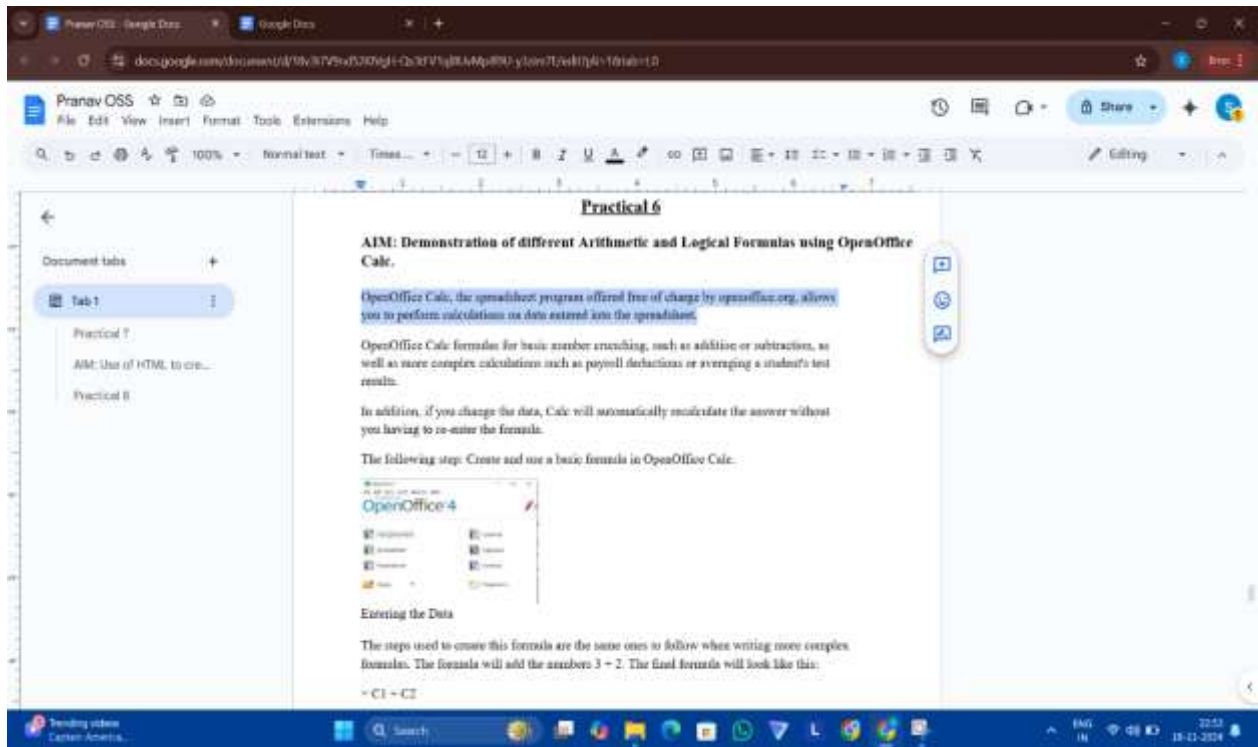
Step 2: Edit and format.

To edit a document:

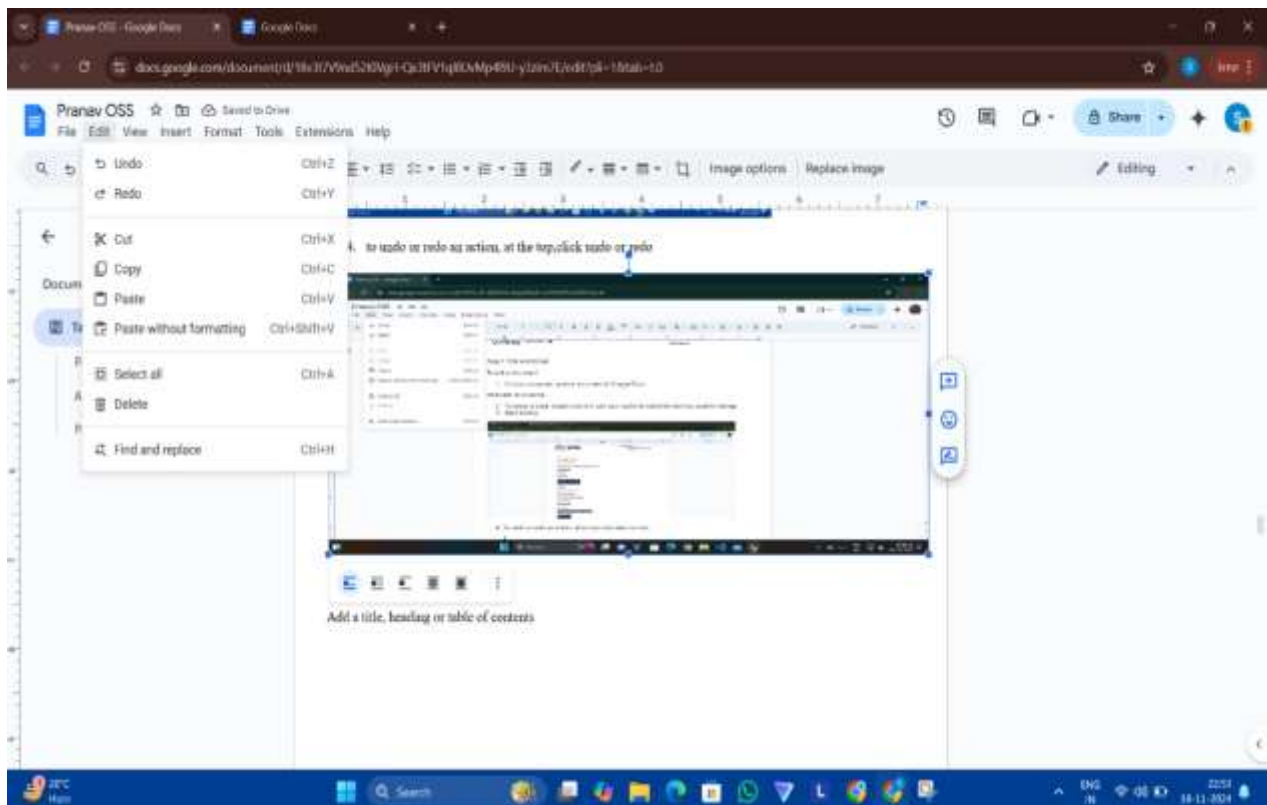
1. On your computer, open a document in Google Docs.



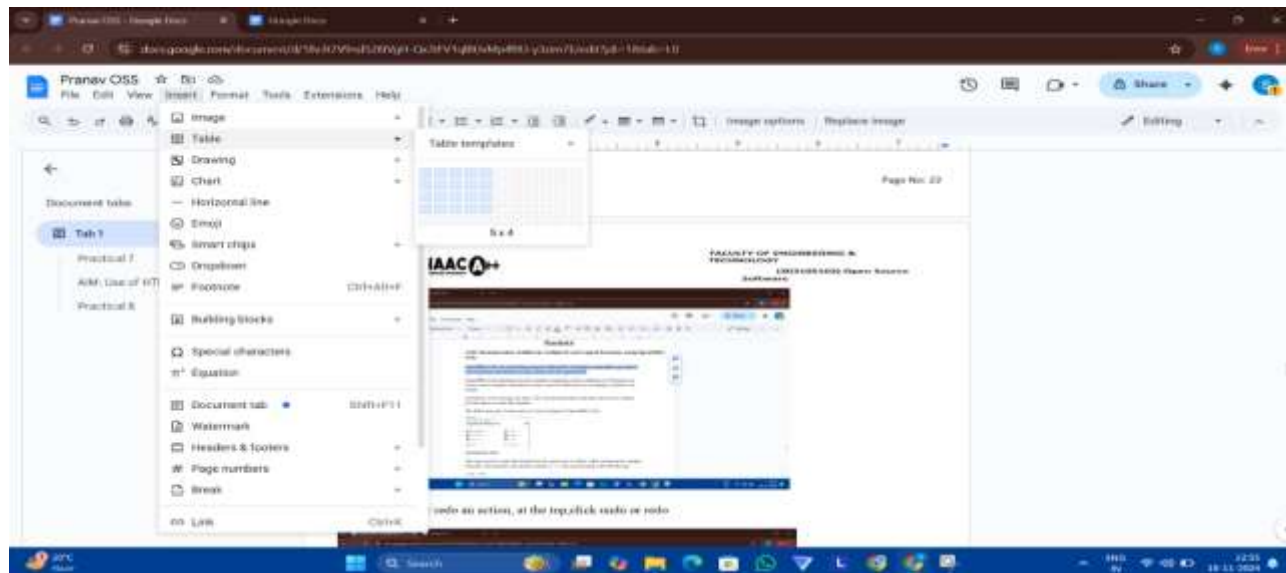
2. To select a word, double click it or use your cursor to select the text you want to change.
3. Start Editing.



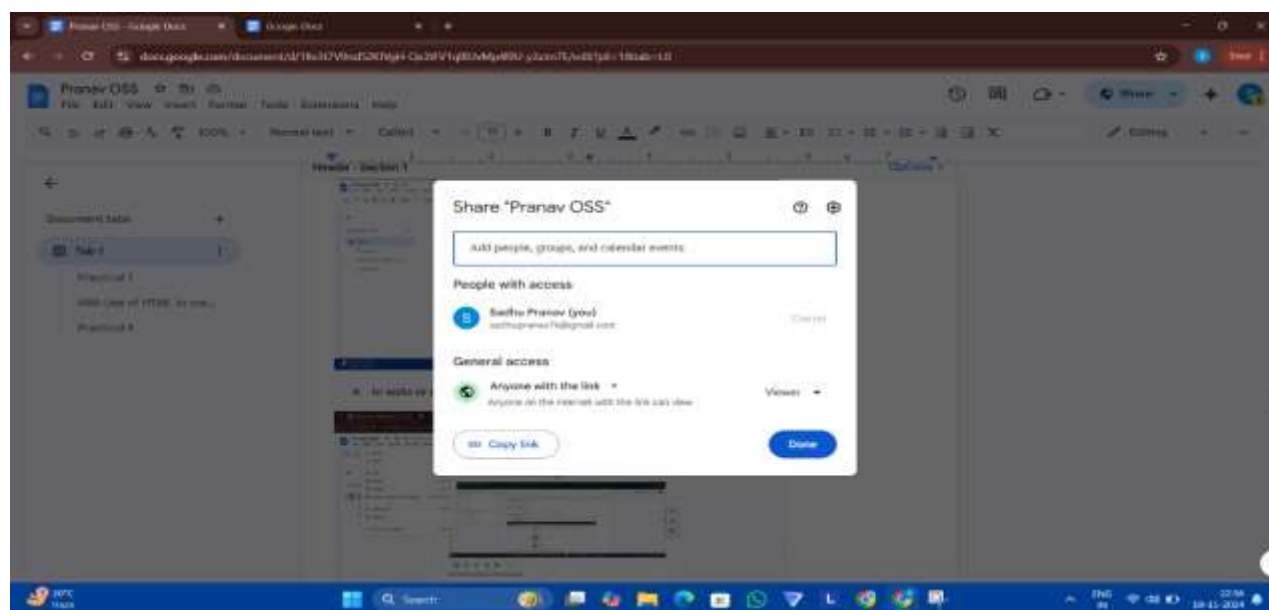
4. to undo or redo an action, at the top, click undo or redo



Add a title, heading or table of contents



Step 3: Share & Work with others



Conclusion:

In conclusion, Google Docs is a powerful tool for creating and editing documents collaboratively online. Its user-friendly interface, real-time collaboration features, and accessibility make it ideal for personal, academic, and professional use.

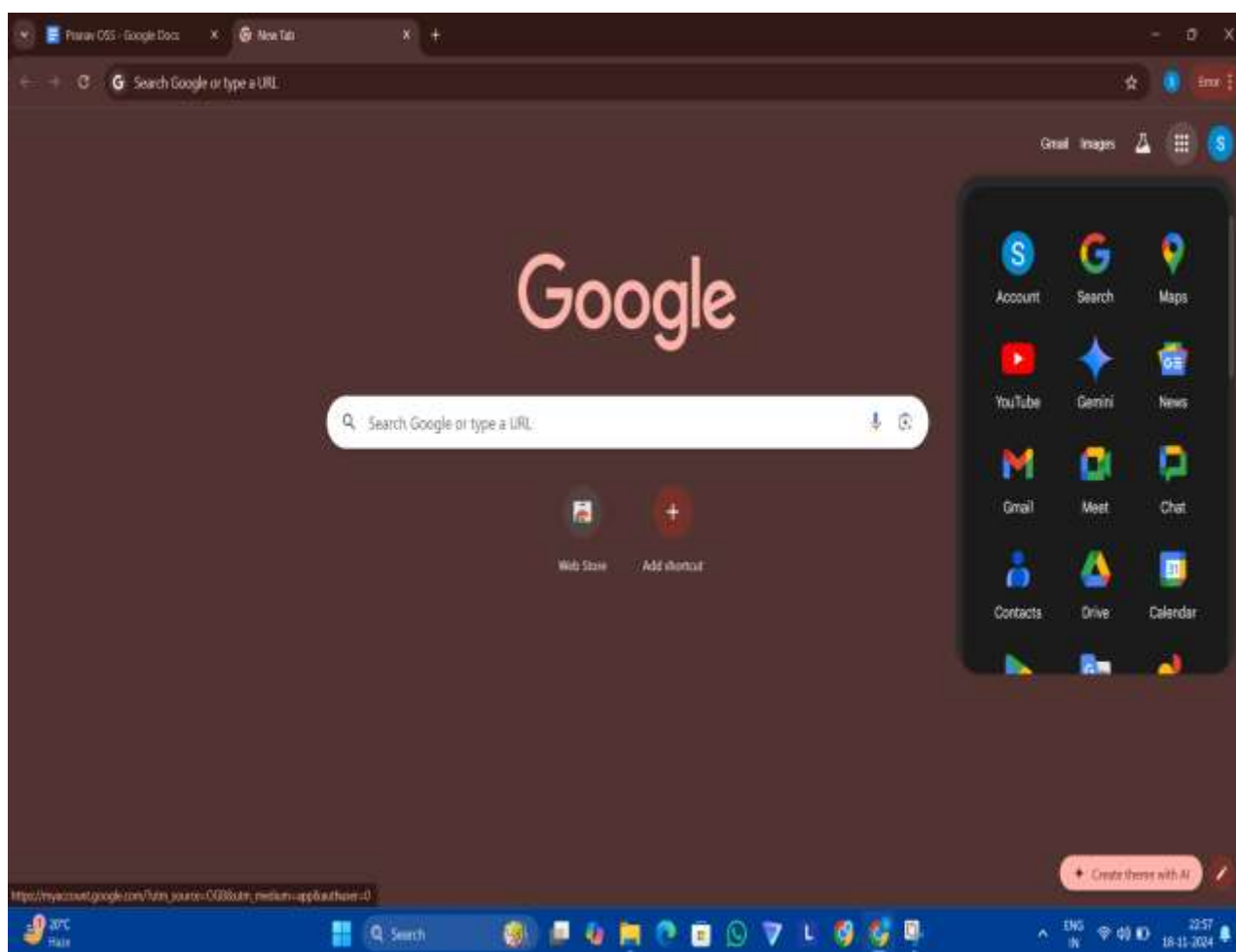
PRACTICAL 5

AIM: Create Presentation using Google Slides.

Go to Google's home page and click on the grid in the upper right hand corner.

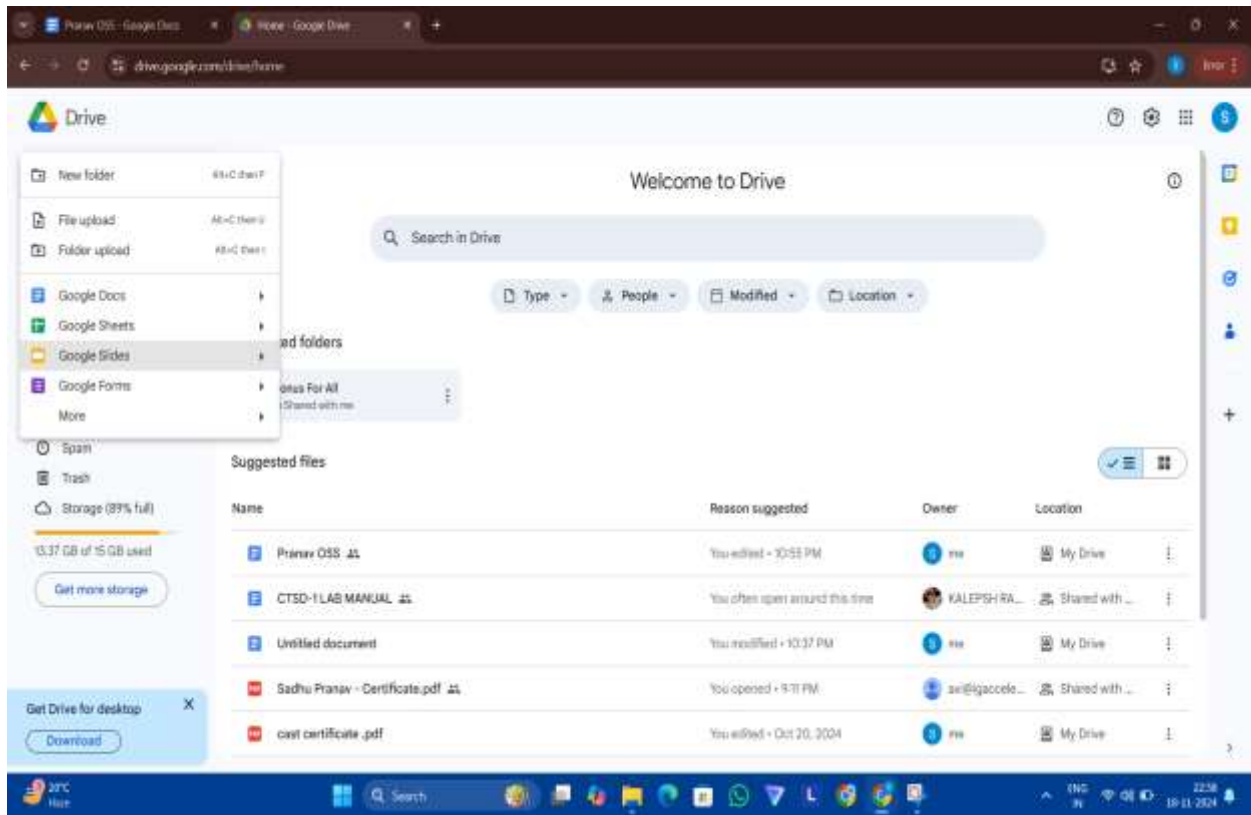
click on the Drive Icon. You will be redirected to a login page if you are signed out, if not you will be taken to your Drive.

You can also just type in <https://slides.google.com>, log in if you haven't already, and will be taken to the slides page.

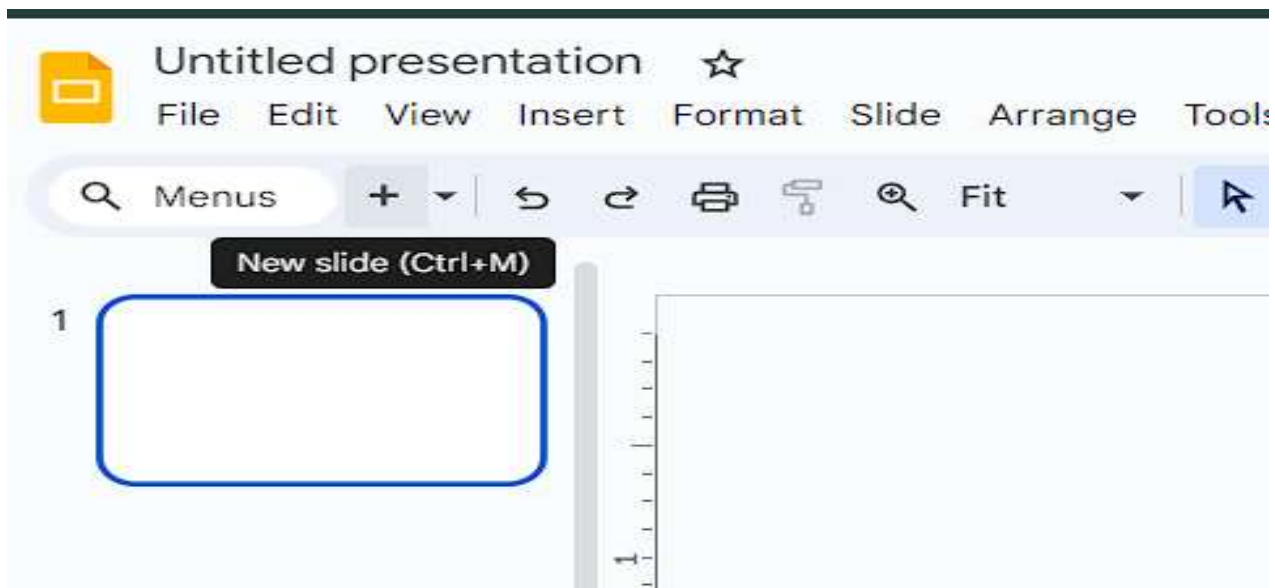


From your Drive, click the blue New button on the left side of the page. Select “Google Slides” from the drop down menu.

For more options, hover over the arrow on the right edge of the Google Slides option, where a smaller drop-down menu will appear. From here you can select to create a presentation from a template or a blank slide.



If you are on the slides page, select an option from the top of the page to create a new slide. You can press the white square with a plus sign for a blank slide, or click one of the templates. Click on the Template Gallery option, where more templates will show up.



Open Source Software



By :- Gummadi Dheeraj

Introduction To Open Source Software

- *Open Source software is open to distribution and modification by anyone in the world - that could be you! Open source projects are not only a way to contribute to the free technology movement, but also a great way to experiment with new languages and frameworks in a welcoming community.*
- *The open source community has created a wide array of projects from software to games to video editors to music to cryptocurrency! The open source community is truly everywhere, contributing to projects by giants like Google or adding to digital public goods for collaborative healthcare efforts worldwide.*

What is Open Source Software?

Open source software (OSS) is software that is available for anyone to use, study, modify, and redistribute. It's typically released under a license that grants users these rights.

❖ Here are some benefits of open source software:

- *Free: OSS is often free to use.*
- *Flexible: Developers can modify the software to suit their needs.*
- *Community-based: OSS encourages collaboration and sharing of ideas.*
- *Diverse perspectives: OSS development can include a variety of perspectives.*
- *Control: Users can check the code to make sure it's not doing anything they don't want it to do.*

Benefits Of Open Source Software



Security

Open source software is more secure because it's publicly accessible, allowing many people to review the code and suggest changes to improve security and functionality.



Transparency

Culture of openness makes new community. Everyone has access to code that allows transparency. So whether we are creating a software or solving a problem then everyone has access to get information about doing best work.



Collaboration

Open source communities can help with projects by providing exposure to coding and access to accomplished developers who can offer assistance.



Reliability

Open source software is monitored by thousands of experts who can quickly fix faults, which helps the software run smoothly.



Customization

User can modify the code as per their need, can fix bugs and can add new features.



Cost Effectiveness

Generally open source software is free to use. So it is very cost effective for businesses, organizations as well as for individuals.

Need Of Open Source Software

1. **Flexibility** : Users can examine and change the code to fit their needs.
2. **Stability** : The source code is publicly distributed, so users can rely on it for long-term projects.
3. **Quality** : Open source software is often considered to have better quality than commercially developed software because more developers review the code.
4. **Control** : Users have more control over the code and how they use it.
5. **No supplier dependence** : Users aren't locked into a relationship with a specific software company.
6. **Technology agility** : Open source software offers multiple ways to solve problems, so users can create capabilities themselves instead of waiting for a vendor.
7. **Cost-savings** : Open source software addresses issues such as expensive license fees.
8. **Attracting talent** : Open source software can help organizations attract better talent and provide opportunities for developers to learn and advance.

Principles Of Open Source Software

- **Redistribution**: The software and its modifications can be redistributed.
- **Source code**: The source code must be included in the program so it can be easily modified and distributed.
- **Integrity**: The software can be used, modified, or redistributed under a free license, as long as the license remains the same.
- **No discrimination**: There should be no discrimination against any person, group, or area of activity.
- **Non-specificity**: The software license should not be specific to a product.
- **No restrictions**: The program can be used for any purpose.
- **Transparency**: All collaboration happens online through open channels.
- **Reliability**: Thousands of experts monitor open-source platforms, which can help fix faults quickly and ensure the software runs smoothly.
- **Flexibility**: Users can modify the source code or add additional modules or extensions.
- **Savings**: Finding existing code that fills specific needs can accelerate a project and free developers to focus on other areas.
- **Release excitement**: A release schedule and roadmap can help users get invested in when updates come out.



Open Source Software Standards

- Open-source standards are generally not a topic of discussing but it play very significant role in very organizations and technologies.
- Open source standards make reference to protocol, technical specifications, data structures and formats with no restrictions.
- These standards promote interoperability, collaboration, and innovation in a particular domain.
- Interoperability, collaboration and innovation are promoted by the standards.
- Here we include some standards of the open source.

Berkeley Software Distribution

Berkeley Software Distribution (BSD) is a group of related open source Unix-like operating systems (OS) with origins in early versions of Research Unix at Bell Labs. FreeBSD is the most popular member.

BSD is configured for internet hosting, web hosting, and hosting many servers on one system. It is the first OS to have added an internet protocol. BSD OSes have a very strongly enforced time-sharing system, which makes them most useful where resources are shared between processes.

As a comparison, the Linux OS is known to be preferred for single-task processes such as supercomputers and desktops. The effective BSD multi-tasking forced-interrupt part of timesharing gets in the way of dedicated single processes.

BSD includes a 'Jails' system which is somewhat analogous to Linux Containers —except with additional security and flexibility in implementation.

Conclusion:

In conclusion, Google Slides provides an intuitive platform for creating, editing, and presenting slideshows online. Its collaborative features, ease of use, and accessibility make it a great tool for both individual and team presentations.

Practical 6

AIM: Demonstration of different Arithmetic and Logical Formulas using OpenOffice Calc.

OpenOffice Calc, the spreadsheet program offered free of charge by openoffice.org, allows you to perform calculations on data entered into the spreadsheet.

OpenOffice Calc formulas for basic number crunching, such as addition or subtraction, as well as more complex calculations such as payroll deductions or averaging a student's test results.

In addition, if you change the data, Calc will automatically recalculate the answer without you having to re-enter the formula.

The following step: Create and use a basic formula in OpenOffice Calc.

Step 1: Launch OpenOffice Calc

Open OpenOffice Calc on your computer.

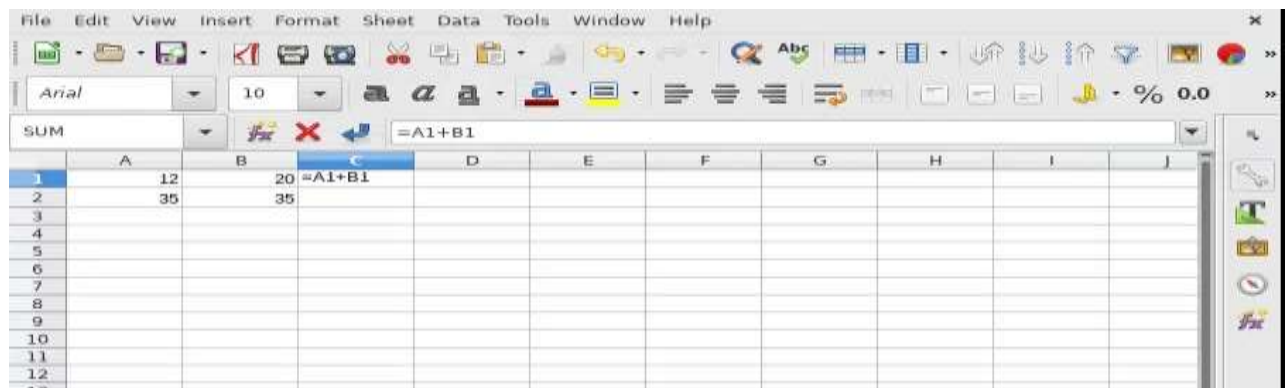


Step 2: Enter Sample Data

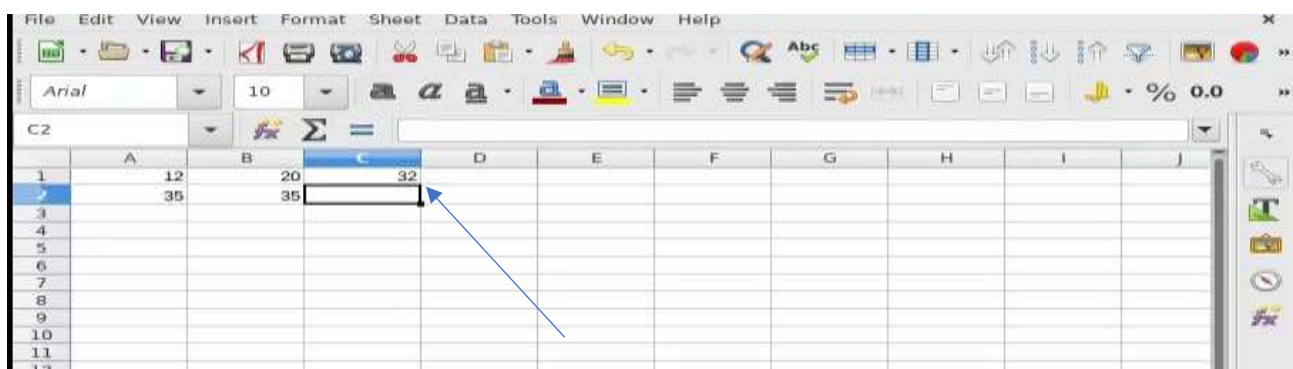
1) Addition:

- In Cell A1, enter 12.
- In Cell B1, enter 20.

- Click on **Cell C1** and type =A1 + B1.

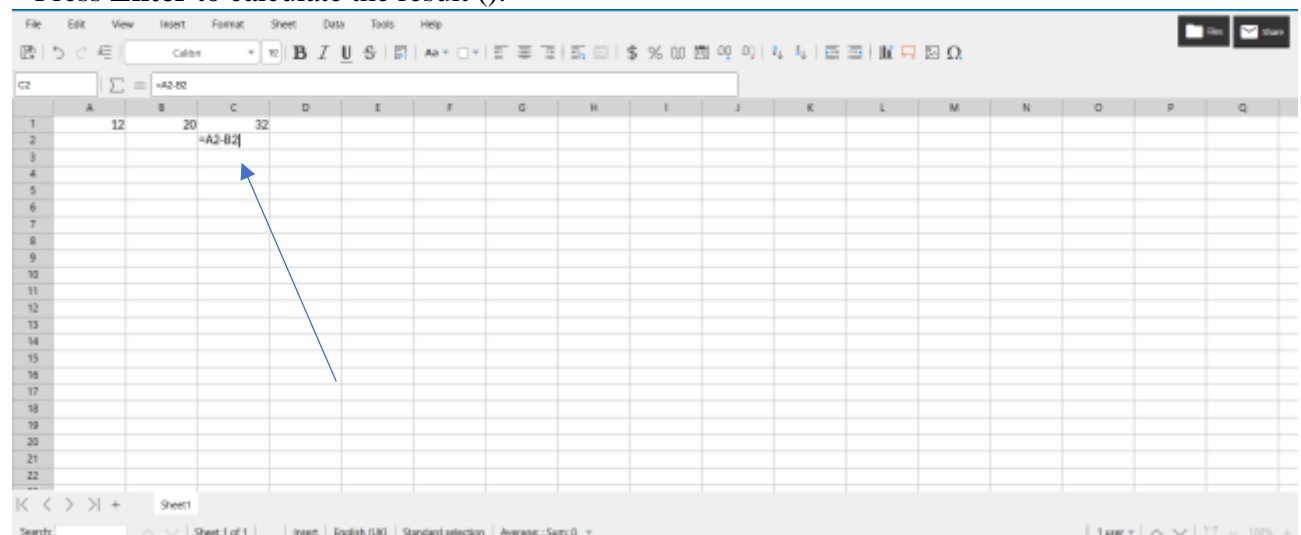


- Press **Enter** to calculate the result (32).



2) Subtraction:

- Click on **Cell C2** and type =A1 - B1.
- Press **Enter** to calculate the result ().



- Press **Enter** to calculate the result (-8).

	A	B	C	D	E	F	G	H
1	12	20	32					
2			-8					
3								
4								
5								
6								
7								
8								
9								
10								

3) Multiplication:

- Click on **Cell C3** and type =A1 * B1.

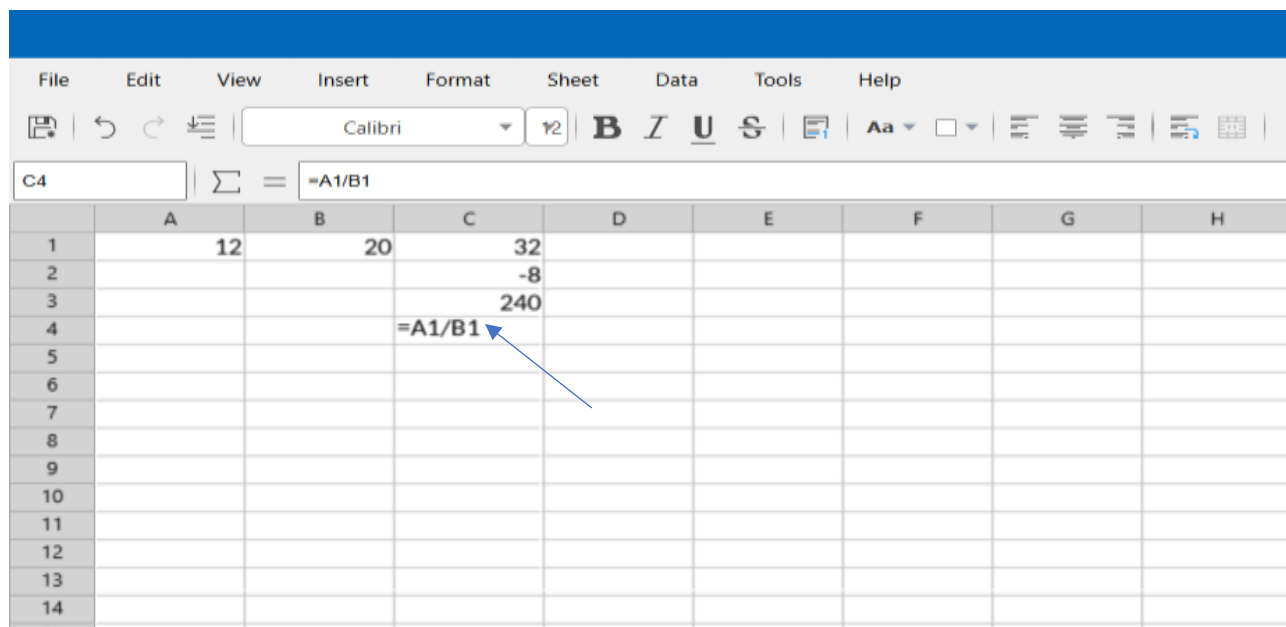
	A	B	C	D	E	F	G
1	12	20	32				
2			-8				
3			=A1*B1				
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							

- Press **Enter** to calculate the result (240).

	A	B	C	D	E
1	12	20	32		
2			-8		
3			240		
4					
5					
6					
7					
8					
9					
10					

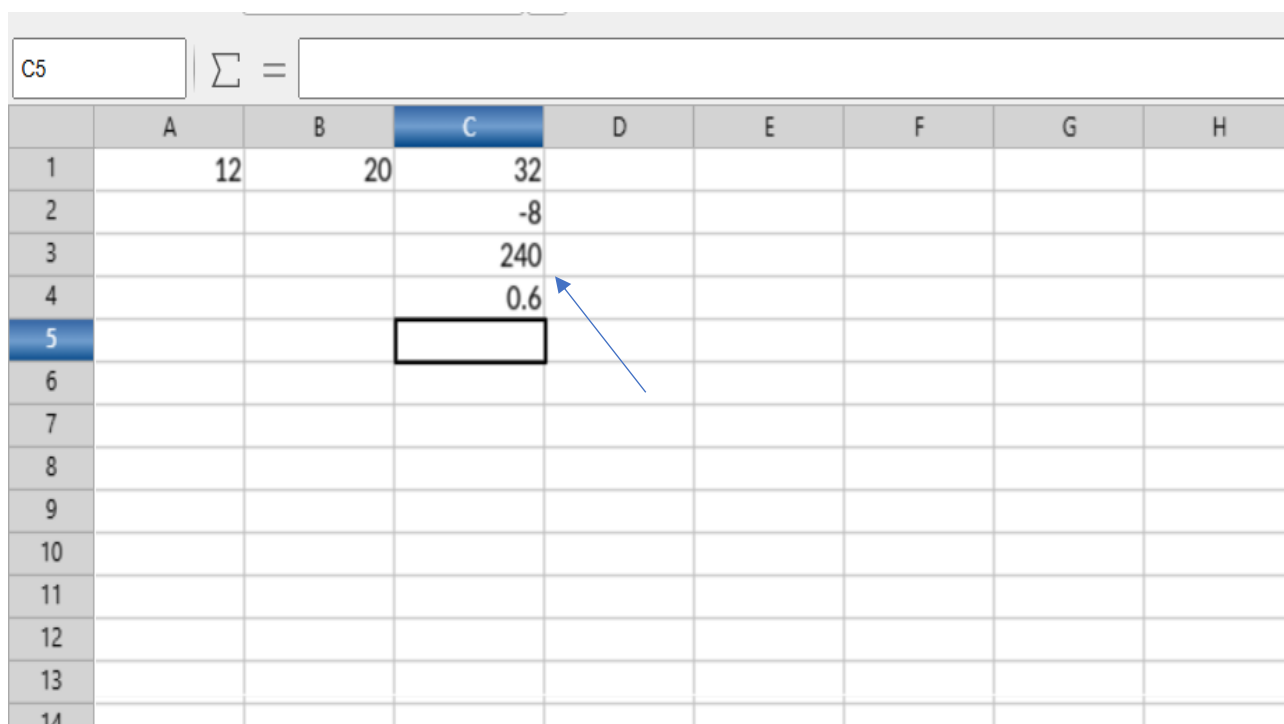
4) Division:

- Click on **Cell C4** and type =A1 / B1.



	A	B	C	D	E	F	G	H
1	12	20	32					
2			-8					
3			240					
4			=A1/B1					
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

- Press **Enter** to calculate the result (0.6).

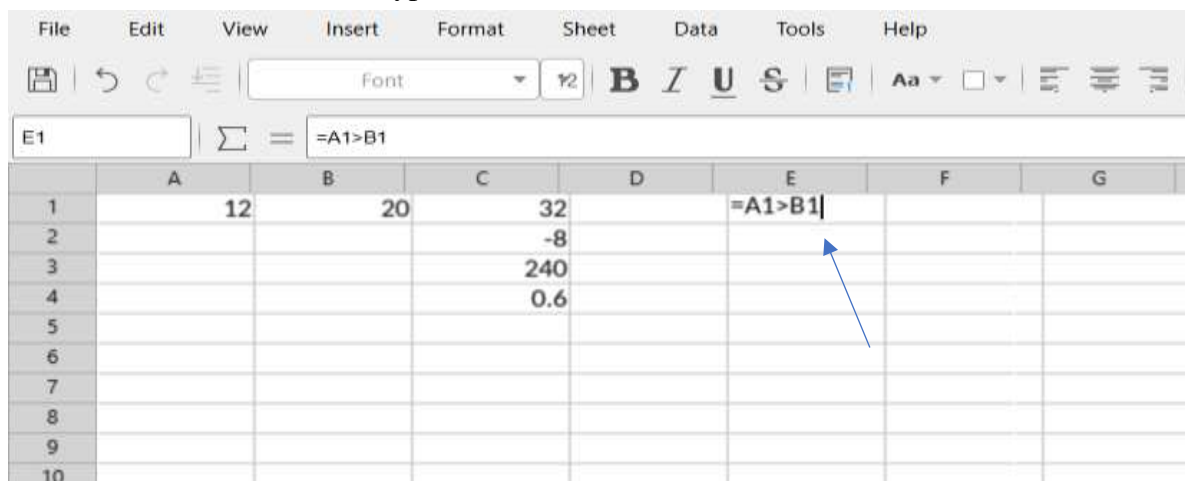


	A	B	C	D	E	F	G	H
1	12	20	32					
2			-8					
3			240					
4			0.6					
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								

Step 4: Add Logical Formulas:

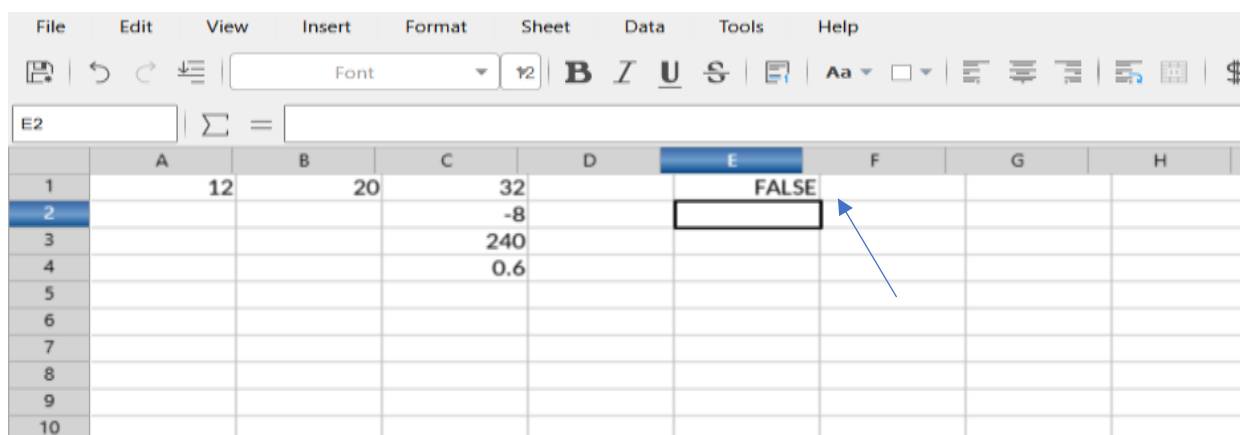
1) **Greater Than Check:**

- Click on **Cell E1** and type `=A1 > B1`.



	A	B	C	D	E	F	G
1	12	20	32		=A1>B1		
2			-8				
3			240				
4			0.6				
5							
6							
7							
8							
9							
10							

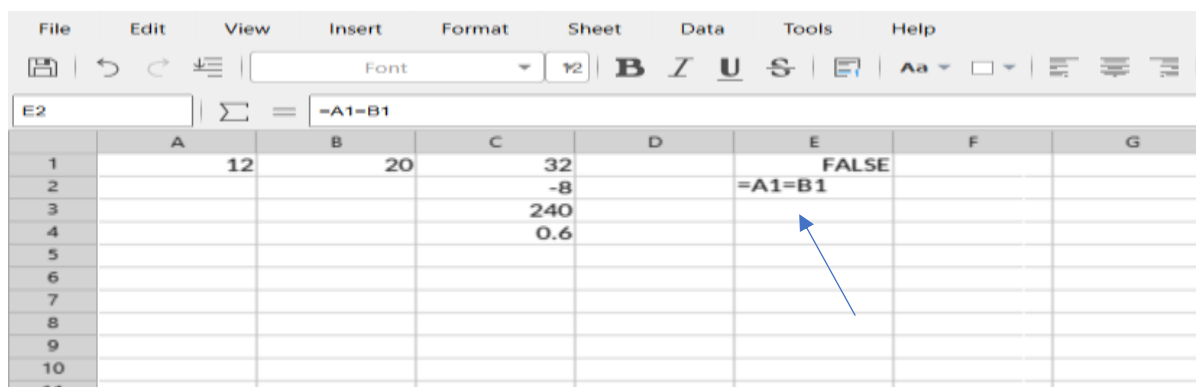
- Press **Enter**. The result will display **FALSE** since 12 is not greater than 20.



	A	B	C	D	E	F	G	H
1	12	20	32		FALSE			
2			-8					
3			240					
4			0.6					
5								
6								
7								
8								
9								
10								

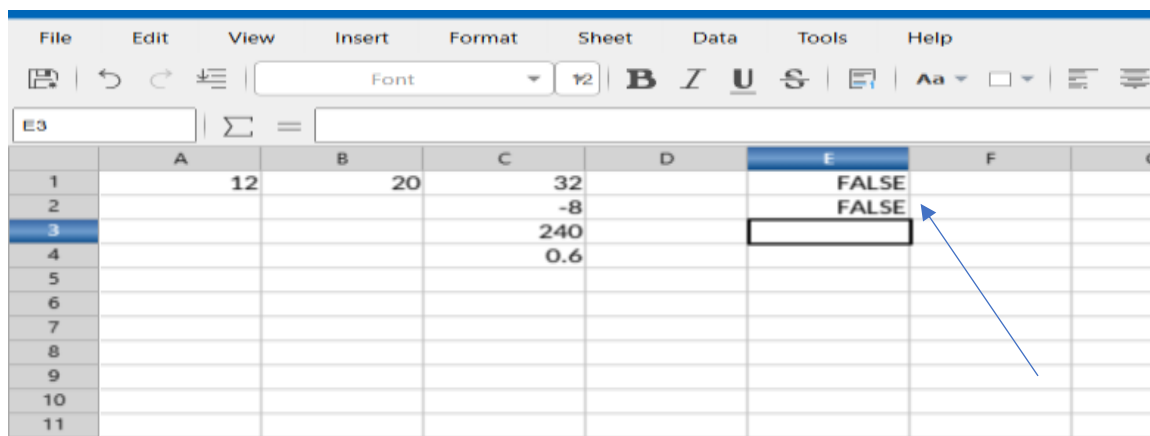
2) **Equal To Check:**

- Click on **Cell E2** and type `=A1 = B1`.



	A	B	C	D	E	F	G
1	12	20	32		FALSE		
2			-8		=A1=B1		
3			240				
4			0.6				
5							
6							
7							
8							
9							
10							
11							

- Press **Enter**. The result will display FALSE.



The screenshot shows the OpenOffice Calc interface. The formula bar at the top displays 'E3' and the formula '= (A3 < B3)'. The spreadsheet grid shows the following data:

	A	B	C	D	E	F	G
1	12	20	32		FALSE		
2			-8		FALSE		
3			240				
4			0.6				
5							
6							
7							
8							
9							
10							
11							

A blue arrow points to cell E3, which contains the text 'FALSE'.

Conclusion:

In conclusion, OpenOffice Calc provides powerful tools for performing both arithmetic and logical operations, enabling efficient data analysis and manipulation. By utilizing functions like addition, subtraction, multiplication, division, and logical tests, users can enhance their spreadsheet work and streamline calculations.

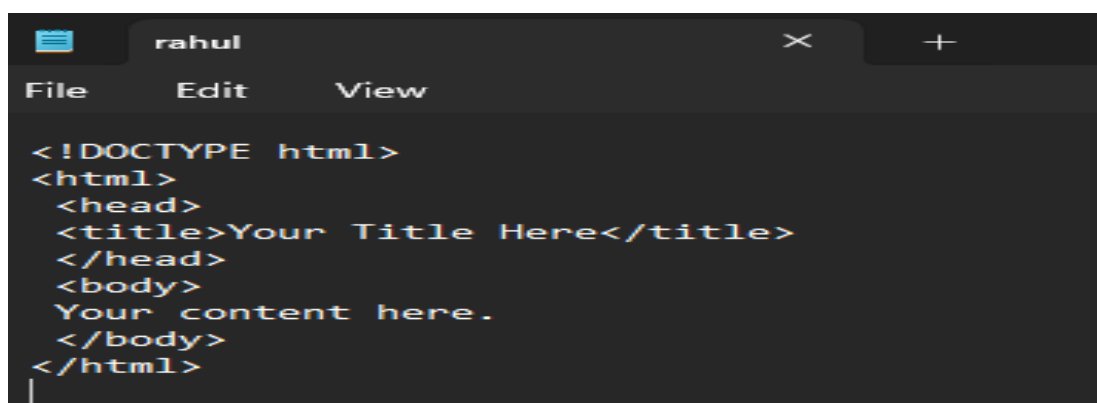
Practical 7

AIM: Use of HTML to create a simple web page.

Step 1: Open a Text Editor (Notepad)

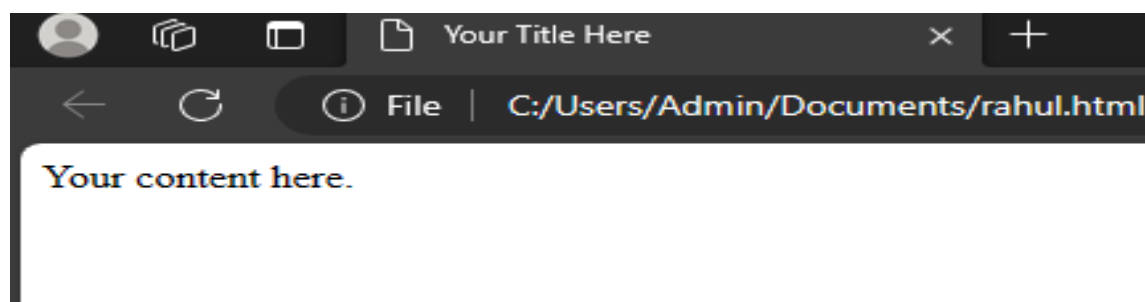
Step 2: Write Some HTML Code

We are now going to add the HTML boilerplate code. This is the code that will allow the browser to correctly display your webpage.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Your Title Here</title>
  </head>
  <body>
    Your content here.
  </body>
</html>
```

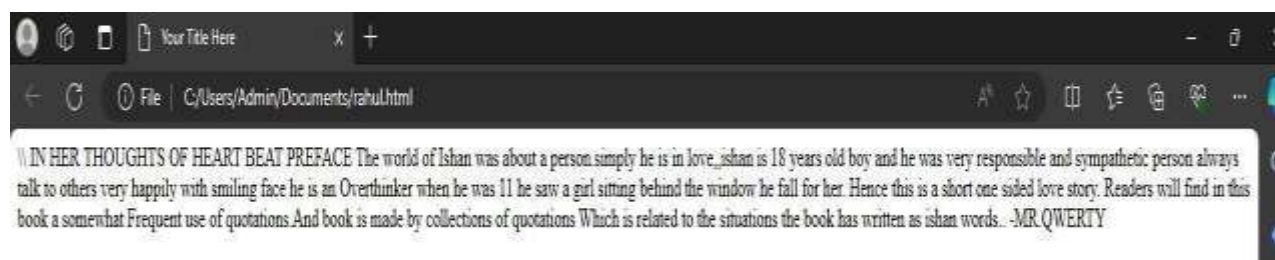
Save the html page by pressing CTRL + S or click on file option then save option. Ensure that you name the file in the following format: “name” then “.html” examples index.html, cooking.html.



Step 3: Add the Text Content

Add the content between the body tags.

Refresh the webpage on the browser. It will look like a blob of text with no paragraphs or headings shown below.



Step 4: Add the HTML

Tags Headings:

First, we shall tackle the headings. Check which part or the text was the main heading.

In our example it is “Insect eating: The ultimate guide to eating crickets”.

Place this text between h1 tags as shown below:

```
<H1>IN HER THOUGHTS OF HEART BEAT</H1> <H2>PREFACE</h2>
```

Save your work and check the result on the browser.



IN HER THOUGHTS OF HEART BEAT

PREFACE

The world of Ishan was about a person. simply he is in love, ishan is 18 years old boy and he was very responsible and sympathetic person always talk to others very happily with smiling face he is an Overthinker when he was 11 he saw a girl sitting behind the window he fall for her. Hence this is a short one sided love story. Readers will find in this book a somewhat Frequent use of quotations. And book is made by collections of quotations Which is related to the situations the book has written as ishan words.. -MR.QWERTY

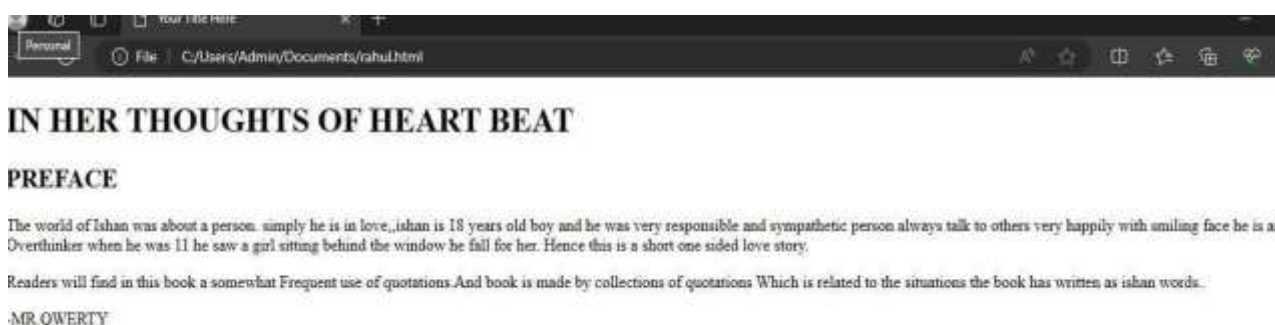
Paragraphs

For the paragraphs of text, place each paragraph of text between the <p></p> html tags.

Example:

```
<H1>IN HER THOUGHTS OF HEART BEAT</H1> <H2>PREFACE</h2>
<P> The world of Ishan was about a person.
simply he is in love,, ishan is 18 years old boy and he was very responsible and sympathetic person always talk to others
very happily with smiling face he is an Overthinker when he was 11 he saw a girl sitting behind the window he fall for her
Hence this is a short one sided love story.</P>
Readers will find in this book a somewhat Frequent use of quotations. And book is made by collections of quotations Which
is related to the situations the book has written as ishan words..<P> -MR.QWERTY</P>
```

OUTPUT:



Step 6: Add a List

Lists make reading a group of things easier on our eyes and brain. Lets add a bullet points type list:

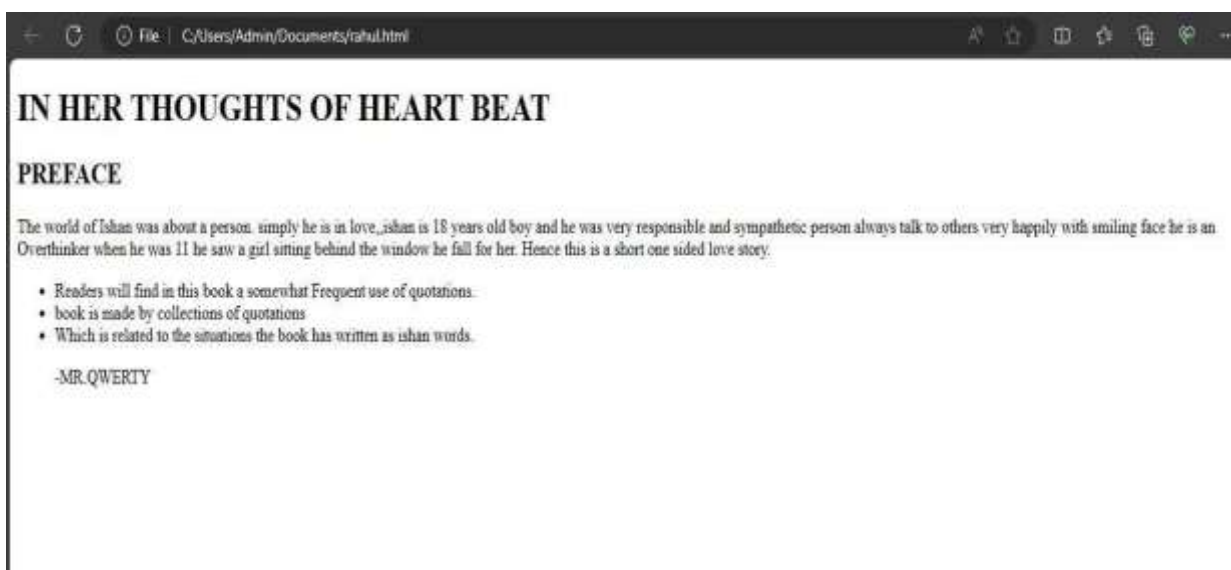
```

Hence this is a short one sided love story.</P>
<ul>
<li> Readers will find in this book a somewhat Frequent use of quotations.</li><li> book is made by collections of quotations</li><li> Which is related to the situations the book has written as ishan words.</li><P> -MR.QWERTY</P>

</body>
</html>

```

Refresh and view the result on the browser.



Step 7: Add Images

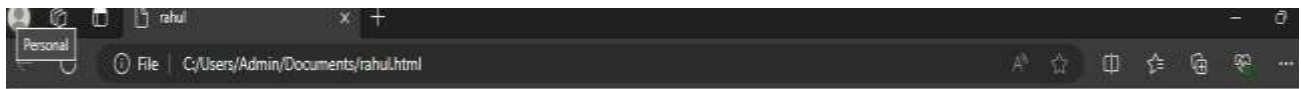
We are going to use the HTML's img tag `` to add

your images on the webpage:

```



```

IN HER THOUGHTS OF HEART BEAT

PREFACE

The world of Ishan was about a person. simply he is in love, ishan is 18 years old boy and he was very responsible and sympathetic person always talk to others very happily with smiling face he is an Overthinker when he was 11 he saw a girl sitting behind the window he fall for her. Hence this is a short one sided love story.

- Readers will find in this book a somewhat Frequent use of quotations.
- book is made by collections of quotations
- Which is related to the situations the book has written as ishan words.

-MR.QWERTY

<https://drive.google.com/file/d/1CTKW78DSyVmn80Jcgn8R9qVZXXLj9EK9/view?usp=drivesdk>

Step 8: Embed a Youtube Video

Search for relevant video on youtube. Once you have found it, click on share button/ link. You will get a popup. Click on embed option

<iframe>

width="310"

height="160.

src="https://www.youtube.com/embed/

BwC4WRKi5QY"

frameborder="0"

allow="accelerometer; autoplay; encrypted-media; gyroscope; picture
in-picture"

allowfullscreen

</iframe>

OUTPUT:



Practical 8

AIM: demonstration of mathml a markup language for describing mathematical notation MathML.

MathML (Mathematical Markup Language) is an XML-based markup language used to represent mathematical notations and formulas.

It is designed to be compatible with XML and HTML, making it suitable for displaying math on webpages or in XML-based documents.

Conceptually, MathML consists of two main strains of markup: Presentation markup is used to display mathematical expressions; and Content markup is used to convey mathematical meaning.

These two strains, along with other external representations, can be combined using parallel markup.

Example:

Here's a simple example of MathML that represents the equation of a line: Xml

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
```

```
<mi>y</mi>
```

```
<mo>=</mo>
```

```
<mi>m</mi>
```

```
<mo>&#x2062;</mo>
```

```
<!-- Invisible times for multiplication -->
```

```
<mi>x</mi>
```

```
<mo>+</mo>
```

```
<mi>b</mi>
```

```
</math>
```

In this example:

<math>: The root element that defines the MathML content.

<mrow>: Represents a row or sequence of mathematical expression

<mi>: Represents a mathematical identifier (in this case, variables).

<mo>: Represents an operator or symbol.

This MathML code represents the equation of a line in slope-intercept form ($y = mx + b$), where "m" and "b" are constants and "x" and "y" are variables.

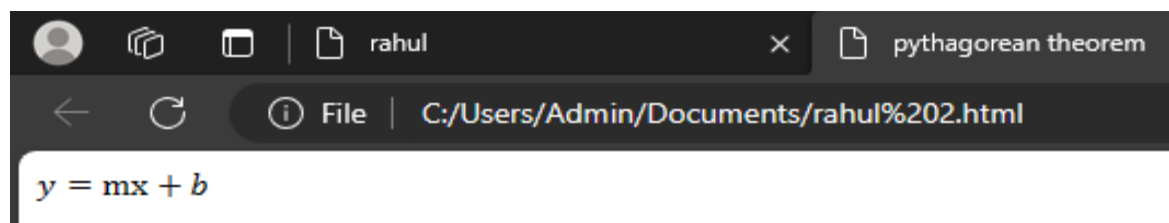
To view this MathML example, you can use a MathML-enabled browser like Firefox, or you can try an online MathML rendering tool.

Copy and paste the MathML code into one of these environments, and you should see the equation of a line displayed.

```
File Edit View

<!DOCTYPE html>
<html>
<head>
<title>pythagorean theorem </title>
</head>
<body>
<math>
<mrow>
<mi>y</mi>
<mo>=</mo>
<mi>mx</mi>
<mo>+</mo>
<mi>b</mi>
</math>
</body>
</html>
|
```

Please note that MathML support may vary in different environments, so make sure you're using a MathML-capable platform to view the example.



Practical 10

AIM: Demonstration of GitHub Facility.

The goal of this demonstration is to familiarize users with GitHub and showcase its various functionalities, which support collaborative software development, version control, and project management. GitHub provides an online platform for hosting and sharing code, tracking changes, and working collaboratively on software projects.

➤ Key Objectives

Understanding GitHub:

Learn what GitHub is and its role in version control and collaboration.

Explore GitHub's integration with Git (the underlying version control system).

➤ Key Features to Demonstrate:

Repository Creation: How to create a new repository to store project files.

Commit and Push: Uploading changes to GitHub and maintaining a history of updates.

Branching and Merging: Managing project features or experiments without affecting the main code.

Pull Requests: Proposing changes and reviewing code collaboratively.

Issues and Discussions: Tracking bugs, feature requests, and facilitating communication.

Actions and Automation: Setting up workflows for CI/CD (Continuous Integration/Continuous Deployment).

Wiki and Documentation: Maintaining project documentation in a structured manner.

Collaboration: Inviting contributors, managing roles, and permissions.

Practical Demonstration:

- Set up a sample project on GitHub.
- Illustrate how to clone the repository locally, make changes, and push updates.
- Show how to use pull requests to review and merge changes.
- Demonstrate how issues and discussions enhance project tracking and collaboration.

Steps for Demonstration

1. Creating a Repository

- Log in to GitHub and click on "New Repository."
- Enter a name, choose visibility (public/private), and add a README file.

2. Cloning the Repository

- Use Git commands (git clone <URL>) to download the repository locally.

3. Making Changes

- Add files or edit existing ones.
- Use Git commands to stage (git add .), commit (git commit -m "message"), and push (git push origin main) changes.

4. Branching and Merging

- Create a new branch (git checkout -b feature-branch).
- Make changes, push the branch, and create a pull request on GitHub.
- Merge the pull request after review.

5. Collaboration Features

- Assign an issue, comment, and close it once resolved.
- Tag contributors, discuss changes, and document progress.

6. Showcase GitHub Actions

- Set up a basic workflow for testing or deploying the application.
- Benefits of GitHub
 - Version Control: Tracks all changes, making it easy to revert to previous states.
 - Collaboration: Facilitates teamwork through branches, pull requests, and discussions.
 - Code Sharing: Allows public or private sharing of repositories.
 - Automation: Simplifies workflows with GitHub Actions.
 - Community: Offers a platform to showcase work and contribute to open-source projects.
- For more details, check out:



Faculty of Engineering & Technology

Subject: Open Source Software LAB

Subject-Code: 303105103

B-TECH: IT Year: 1st Sem: 1st

Conclusion:

In conclusion, GitHub is an essential platform for developers, offering seamless version control, collaboration, and project management tools. It enhances productivity with features like repositories, pull requests, and automation, making it a vital tool for modern software development.