

# **Source Code Management**

**Slot: L15-L16**

**Name: Siddesh P.M**

**SEN No.: A86605224309**

**Faculty: Dr Monit Kapoor**

# INDEX

| <b>S. No.</b> | <b>Lab Session Title</b>         | <b>Page No.</b> |
|---------------|----------------------------------|-----------------|
| <b>1</b>      | <b>Git Fundamentals</b>          | <b>3</b>        |
| <b>2</b>      | <b>Installing Git on Windows</b> | <b>4</b>        |
| <b>3</b>      | <b>Basic CLI Commands</b>        | <b>5</b>        |
| <b>4</b>      | <b>Vim Text Editor</b>           | <b>7</b>        |
| <b>5</b>      | <b>Git Commands</b>              | <b>12</b>       |

# Lab Session 1: Git Fundamentals

## Computer

A **computer** is any device capable of performing calculations, whether they are logical or mathematical.

## Program/Code

A **program** (or **code**) is a set of instructions, often organized as an algorithm, that directs a computer to perform a specific task.

## Need for Managing Source Code

Modern applications, such as Spotify, consist of multiple programs working together on both the frontend and backend to deliver smooth user experience.

Regular updates are essential for:

- **Fixing Bugs:** Quickly resolving errors that may occur.
- **Improving UI/UX:** Enhancing the user interface and overall experience.
- **Optimizing Performance:** Addressing and refining issues for better performance.

For programmers, effective management of source code is crucial because:

- It ensures that all files remain in context throughout the lifecycle of the program.
- It facilitates collaboration, allowing multiple developers to work together on a shared codebase.

## Tools for Source Code Management

### 1. Git:

A version control system that runs locally on your computer. Git helps track changes and manage versions of your project.

### 2. GitHub:

A global, cloud-based platform that hosts Git repositories, enabling developers to share, collaborate, and contribute to projects from anywhere in the world.

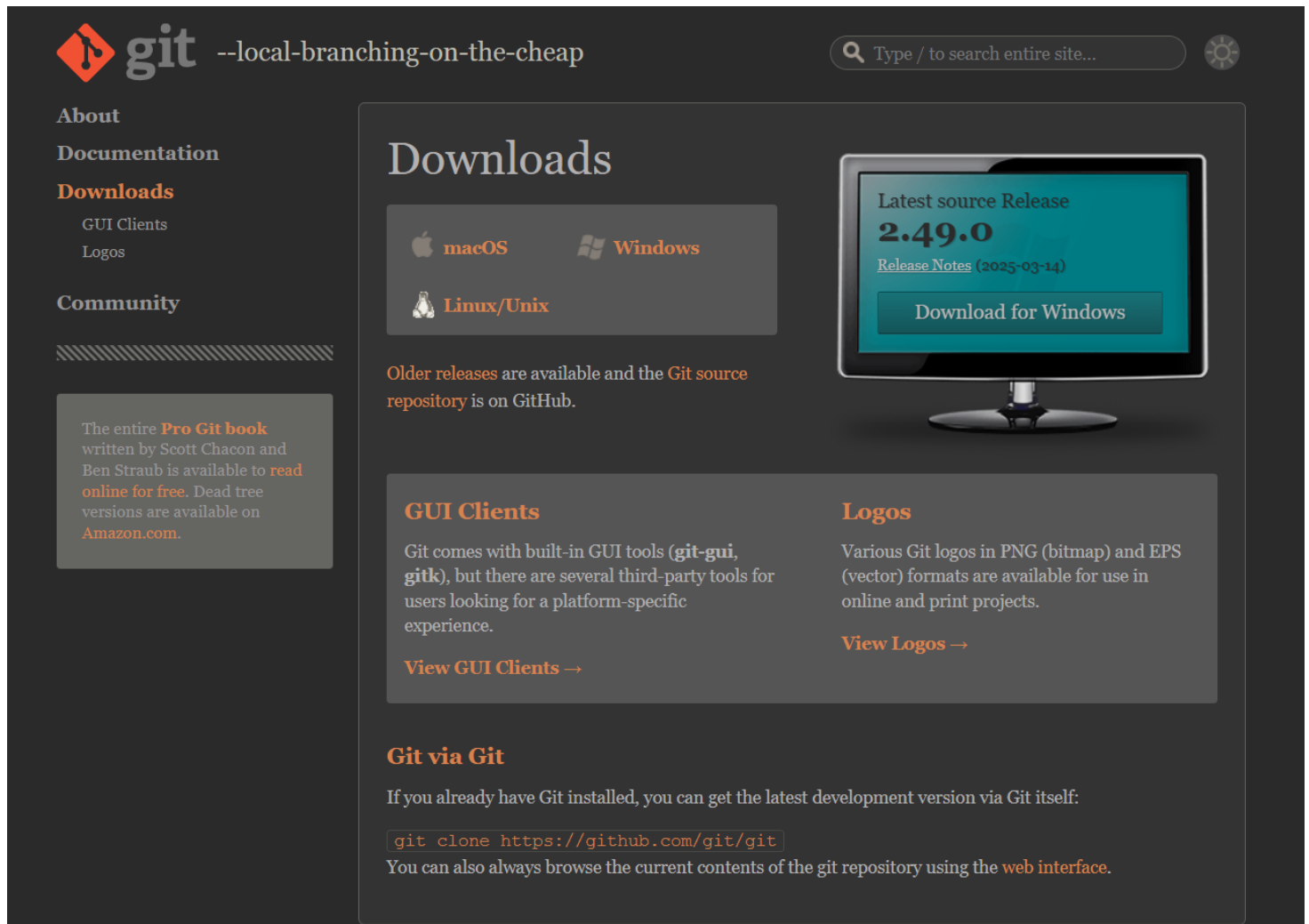
## Version

A **version** in version control represents a snapshot of your project at a specific moment in time. This snapshot allows you to review, revert, or compare changes made throughout the development process.

# Lab Practical 1

## 1. Installing Git in Windows

Step 1: Visit section 1.5 of pro git document and navigate to Windows section



Step 2: Verify Git Installation:

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ git --version  
git version 2.49.0.windows.1
```

## 2. Basic CLI Commands

### 1) Command: pwd

**Description:** Prints the directory the user is working in.

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ pwd  
/c/Users/Shreenidhi
```

### 2) Command: ls

**Description:** Lists all files and directories in the current directory.

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ ls  
AppData/  
'Application Data'@  
Contacts/  
Cookies@  
Documents/  
Downloads/  
Favorites/  
Links/  
'Local Settings'@  
Music/  
'My Documents'@  
NTUSER.DAT  
NTUSER.DAT{db861482-b7d6-11ef-96d3-b37a9bf28160}.TM.blf  
NTUSER.DAT{db861482-b7d6-11ef-96d3-b37a9bf28160}.TMContainer00000000000000000000  
1.regtrans-ms  
NTUSER.DAT{db861482-b7d6-11ef-96d3-b37a9bf28160}.TMContainer00000000000000000000  
2.regtrans-ms  
NetHood@  
OneDrive/  
PrintHood@  
Recent@  
'Saved Games'/  
Searches/  
SendTo@  
'Start Menu'@  
Templates@  
Videos/  
ntuser.dat.LOG1  
ntuser.dat.LOG2  
ntuser.ini
```

### 3. Command: date

**Description:** shows the current date and time in a standard format

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ date  
Wed Jun 4 22:39:34 IST 2025
```

## 4. Command: clear

Description: The `clear` command in the CLI is used to clear all the current text and output displayed in the terminal window.

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ date  
Wed Jun  4 22:39:34 IST 2025  
  
Shreenidhi@SHRINIDHI MINGW64 ~  
$ time  
  
real    0m0.001s  
user    0m0.000s  
sys     0m0.000s
```

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$
```

## 5. Command: time

Description: The `time` command in the CLI is used to measure the execution time of a command or program.

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ time  
  
real    0m0.001s  
user    0m0.000s  
sys     0m0.000s
```

## 6. Command: cd 'Directory'

**Description:** Changes the current working directory to the desired directory.

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ cd documents  
  
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ |
```

## 7. Command: cd ..

Description: Goes back to the previous directory.

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ cd documents  
  
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ cd..  
bash: cd..: command not found  
  
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ cd ..  
  
Shreenidhi@SHRINIDHI MINGW64 ~  
$
```

## 8. Command: mkdir

**Description:** To create a new directory.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ mkdir trail  
  
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ ls  
'My Music'@  'My Pictures'@  'My Videos'@  trail/
```

## 9. Command: rmdir

**Description:** To delete a directory

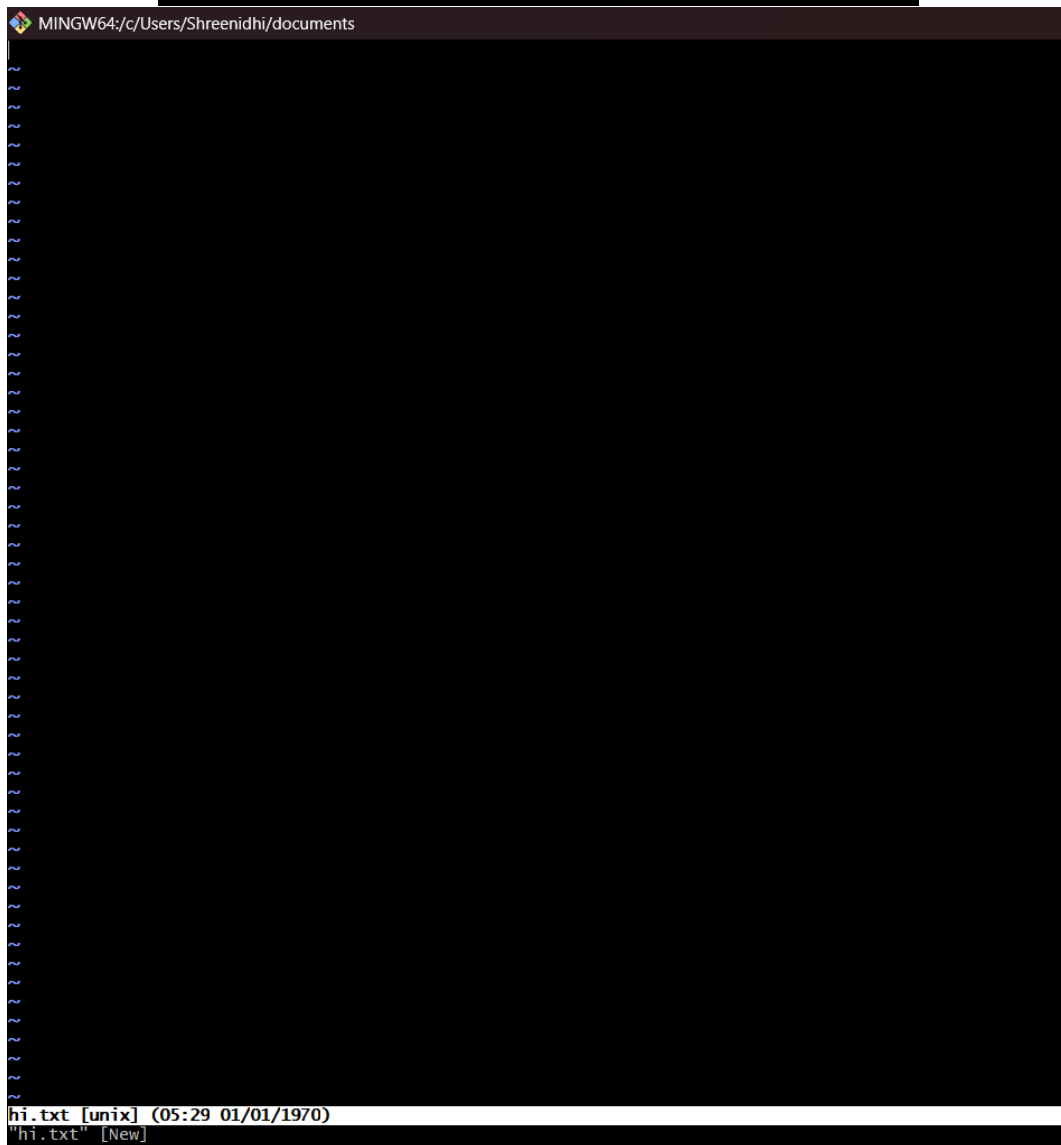
```
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ rmdir trail  
  
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ ls  
'My Music'@  'My Pictures'@  'My Videos'@
```

# 3. Vim Text Editor

## 1) Command: vi hi.txt

**Description:** Opens (or creates) the file `hi.txt` in the Vim text editor.

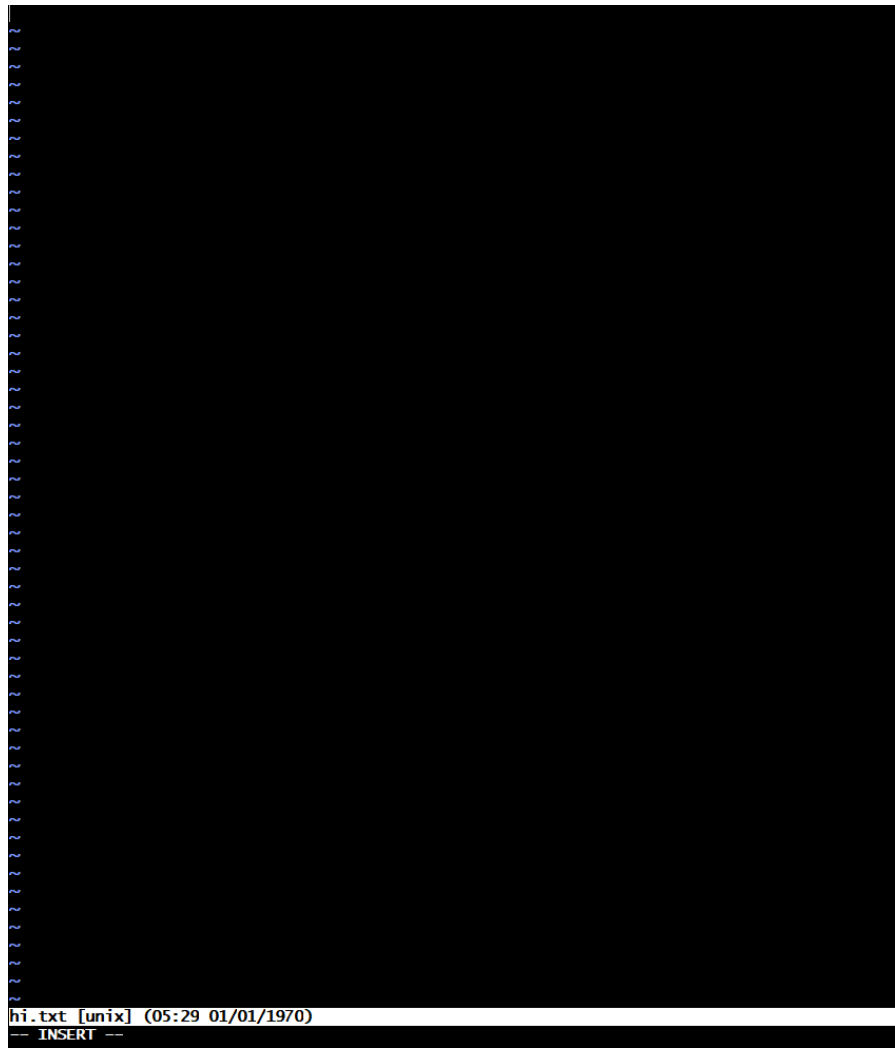
```
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ vi hi.txt
```



## 2) Command: i (Insert Mode)

**Description:** Enters insert mode in Vim to allow text input.







The screenshot shows a terminal window titled "MINGW64:/c/Users/Shreenidhi/documents". Inside the terminal, a text editor is open, displaying the first three lines of a file: "First Line", "Second Line", and "Third Line". The editor is in insert mode, indicated by the status bar at the bottom which reads "hi.txt[+] [unix] (05:29 01/01/1970) -- INSERT --". The left margin of the editor contains a vertical column of tilde (~) characters, suggesting a large number of lines in the file.

### 3) Command: esc

Description: Used to exit insert mode



#### 4) **Command:** :wq

**Description:** Saves the changes and exits the Vim editor.

```
~  
~  
~  
~  
hi.txt[+] [unix] (05:29 01/01/1970)  
:wq|
```

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ vi hi.txt  
  
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ ls  
'My Music'@  'My Pictures'@  'My Videos'@  hi.txt
```

## 4. Git Commands

### 1. Command: git - - version

Description: The `git --version` command is used to check the installed version of Git on your system.

```
Shreenidhi@SHRINIDHI MINGW64 ~  
$ git --version  
git version 2.49.0.windows.1
```

### 2. Command: git init

**Description:** Initializes a new Git repository in the current directory.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents  
$ git init  
Initialized empty Git repository in C:/Users/Shreenidhi/Documents/.git/
```

### 3. Command: git status

**Description:** Displays the current status of the working directory and staging area.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git status
warning: could not open directory 'My Music/': Permission denied
warning: could not open directory 'My Pictures/': Permission denied
warning: could not open directory 'My Videos/': Permission denied
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hi.txt

nothing added to commit but untracked files present (use "git add" to track)
```

### 4. Command: git add Test.c

**Description:** Add Test.c to the staging area in preparation for a commit.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git add hi.txt
warning: in the working copy of 'hi.txt', LF will be replaced by CRLF the next time Git touches it
```

### 5. Command: git commit -m "add file one"

**Description:** Commits the stage changes with the message "add file one".

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git commit -m "Add file one"
[master (root-commit) 9cac84a] Add file one
1 file changed, 3 insertions(+)
create mode 100644 hi.txt
```

### 6. Command: git log

**Description:** Display the commit history of the repository.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git log
commit 9cac84a919d26b8f810589e1778c5e95299e2e1c (HEAD -> master)
Author: siddesh996 <siddesh.m1@s.amity.edu>
Date: Wed Jun 4 22:49:36 2025 +0530

    Add file one
```

## 7. Command: git clone

**Description:** To obtain a copy of an existing Git repository.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git clone https://github.com/AsmSafone/MusicPlayer
Cloning into 'MusicPlayer'...
remote: Enumerating objects: 266, done.
remote: Counting objects: 100% (118/118), done.
remote: Compressing objects: 100% (35/35), done.
Receiving objects: 100% (266/266), 1.84 MiB | 3.34 MiB/s, done.
remote: Total 266 (delta 98), reused 83 (delta 83), pack-reused 148 (from 2)
Resolving deltas: 100% (146/146), done.
```

## 8. Command: git log --oneline

**Description:** For generating shorter commit ID.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git log --oneline
9cac84a (HEAD -> master) Add file one
```

## 9. Command: git diff

**Description:** To compare two files.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git diff b46a555 521a0d3
fatal: ambiguous argument 'b46a555': unknown revision or path not in the working
tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
```

## 10. Command: git remote add "Variable"

**Description:** To connect with the Users GitHub account.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git remote add origin https://github.com/siddesh996/Lab-File.git
```

## 11. Command: git remote

**Description:** To check the status of the repositories connected with the Users account.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git remote
origin
```

## 12. Command: git push -u “Variable” master

**Description:** To push all the files to the Users account.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 235 bytes | 235.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/siddesh996/Lab-File.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

## 13. Command: git merge “File\_Name” -m “comment”

**Description:** To merge a branch with main branch.

```
Shreenidhi@SHRINIDHI MINGW64 ~/documents (main)
$ git merge master -m "Merging siddu"
Already up to date.
```