Assignment No. 1

Aim : Git Installation & Setup

a. Install Git on your system.
b. Configure Git with your name and email.
c. Check Git version and setup verification.

Objective : To install Git and configure it with a username and email ID for effective version control usage.

Git - Version Control System

Git is a distributed version control system (DVCS) used to manage and track changes in source code during software development. It allows developers to record the history of a project, collaborate with others, and revert to previous versions of files when needed.

Git works by creating snapshots of the project instead of storing file differences. Each snapshot represents the complete state of the project at a specific point in time, ensuring fast performance and reliable version tracking.

One of the key features of Git is its distributed nature. Every developer has a full copy of the repository on their local system, including the entire history. This allows users to work offline, commit changes locally, and synchronize with remote repositories when an internet connection is available.

Git provides powerful tools such as:

- Branching and merging to work on features independently
- Commit history to track who made changes and when
- Reverting and rollback to recover from mistakes
- Collaboration support with remote repositories like GitHub

Because of its speed, flexibility, and reliability, Git is widely used in software development for maintaining code quality, teamwork, and project version control.

Git Workflow

1. Working Directory

- This is the folder where the developer is currently working.
- Files are created, modified, or deleted here.
- At this stage, Git does not track changes automatically.

2. Staging Area

- The staging area is an intermediate area between working directory and repository.
- Files are added here before making a commit.

- It allows the developer to select specific files to be committed.

3. Local Repository

- The local repository stores permanent snapshots (commits) of the project.
- After committing, changes are saved locally with a unique commit ID.
- Every developer has their own local repository.

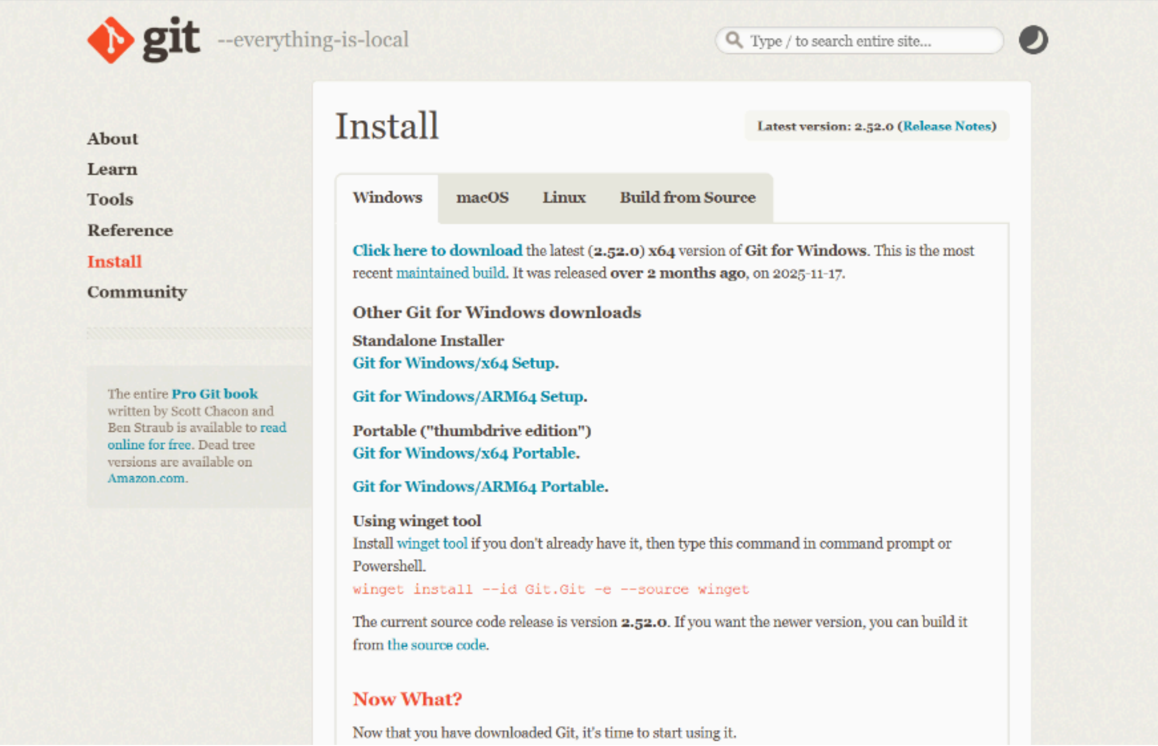
4. Remote Repository

- The remote repository is stored on platforms like GitHub.
- It is used for collaboration among multiple developers.
- Developers share their changes by pushing commits.

Git installation

Step 1: Download Git

1. Open a browser
2. Go to https://git-scm.com
3. Click Download for Windows
4. The download starts automatically



Install

Latest version: 2.52.0 (Release Notes)

Click here to download the latest (**2.52.0**) **x64** version of **Git for Windows**. This is the most recent maintained build. It was released **over 2 months ago**, on 2025-11-17.

Other Git for Windows downloads

Standalone Installer
Git for Windows/x64 Setup.

Git for Windows/ARM64 Setup.

Portable ("thumbdrive edition")
Git for Windows/x64 Portable.

Git for Windows/ARM64 Portable.

Using winget tool
Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.52.0**. If you want the newer version, you can build it from the source code.

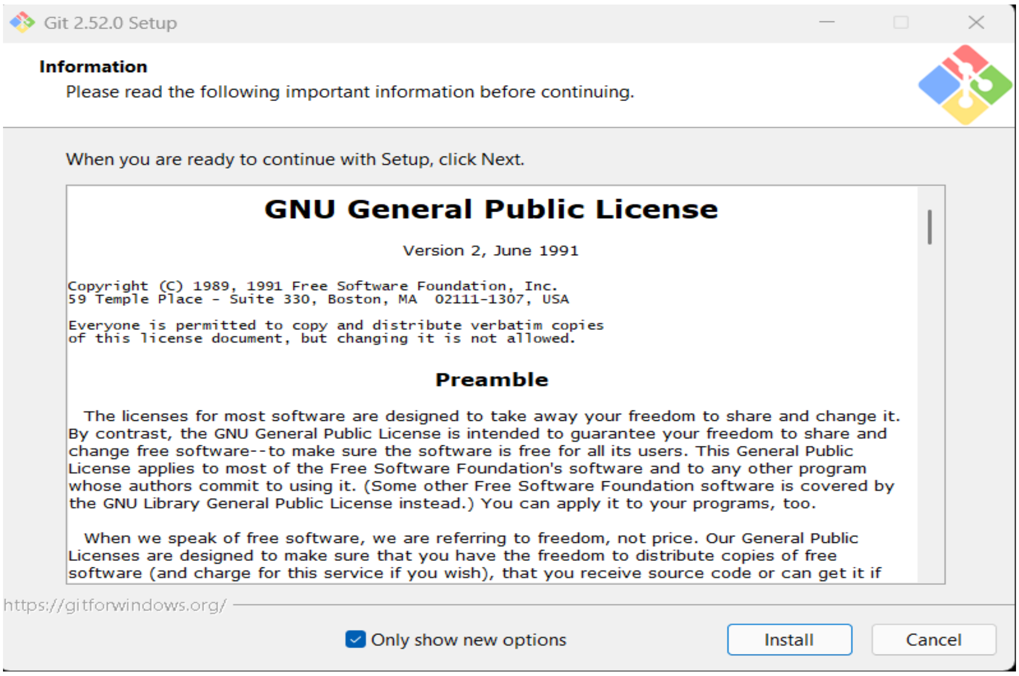
Now What?

Now that you have downloaded Git, it's time to start using it.

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Step 2: Run the Installer

1. Open the downloaded .exe file
2. Click Next on the welcome screen



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

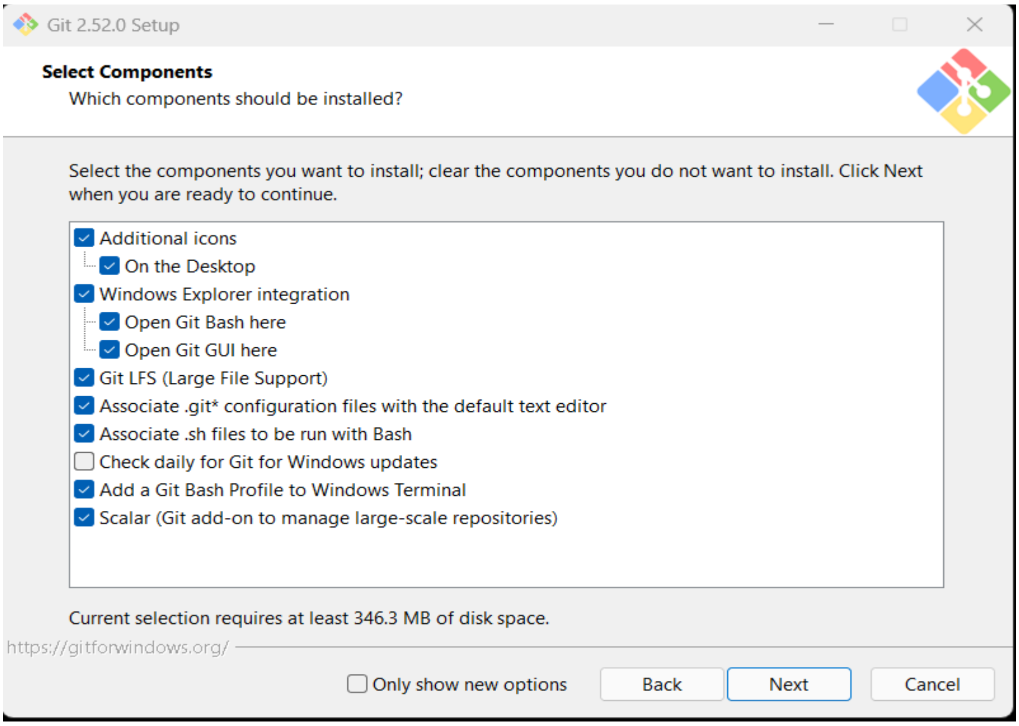
When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if

https://gitforwindows.org/

☑ Only show new options

Step 3: Installation

- Select destination location → Next
- Select components → Next



Select Components

Which components should be installed?

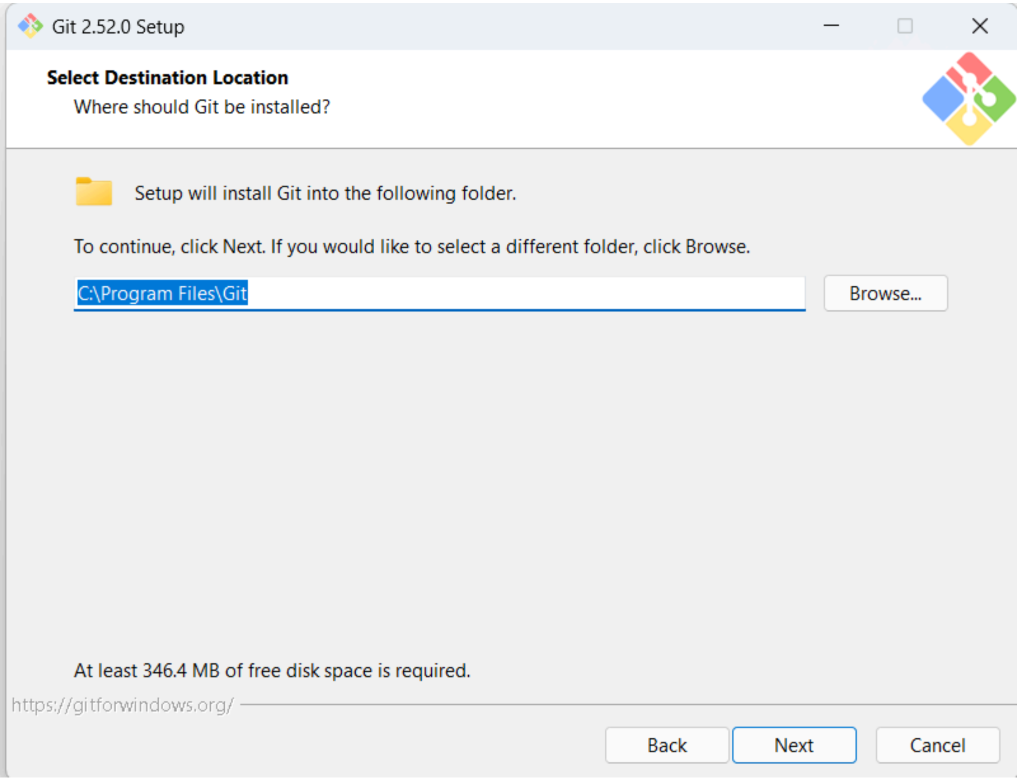
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Additional icons
  - ☑ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Open Git Bash here
  - ☑ Open Git GUI here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Check daily for Git for Windows updates
- ☑ Add a Git Bash Profile to Windows Terminal
- ☑ Scalar (Git add-on to manage large-scale repositories)

Current selection requires at least 346.3 MB of disk space.

https://gitforwindows.org/

□ Only show new options



Select Destination Location

Where should Git be installed?

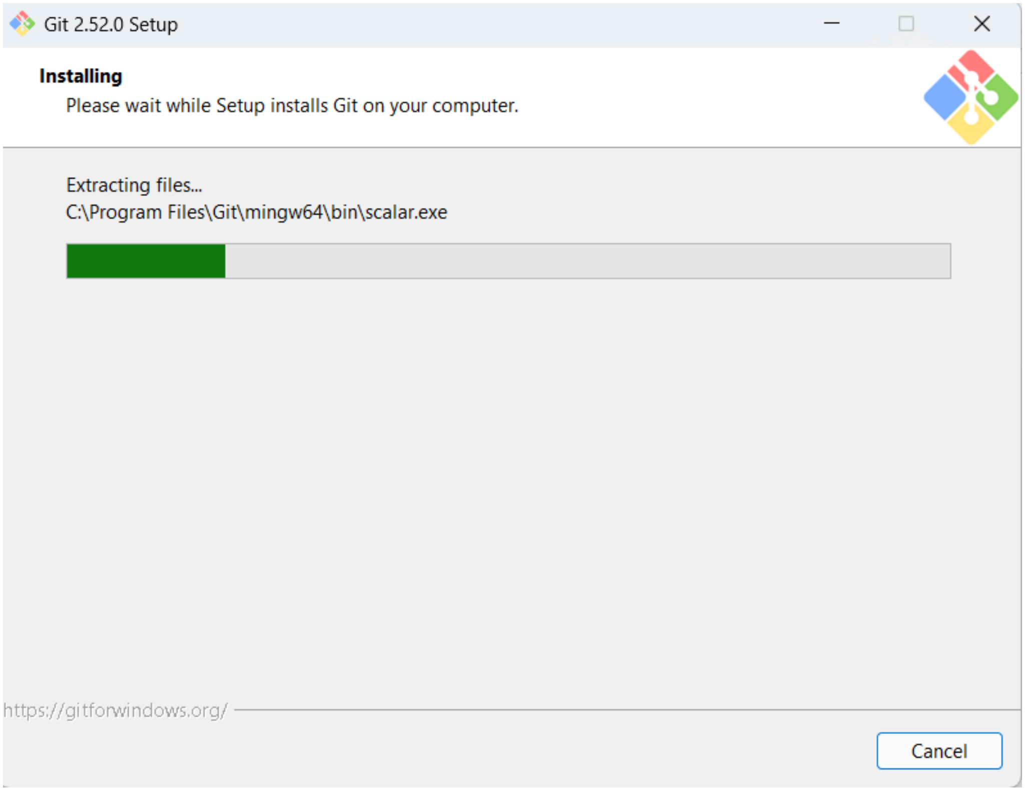
Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 346.4 MB of free disk space is required.

https://gitforwindows.org/



Installing

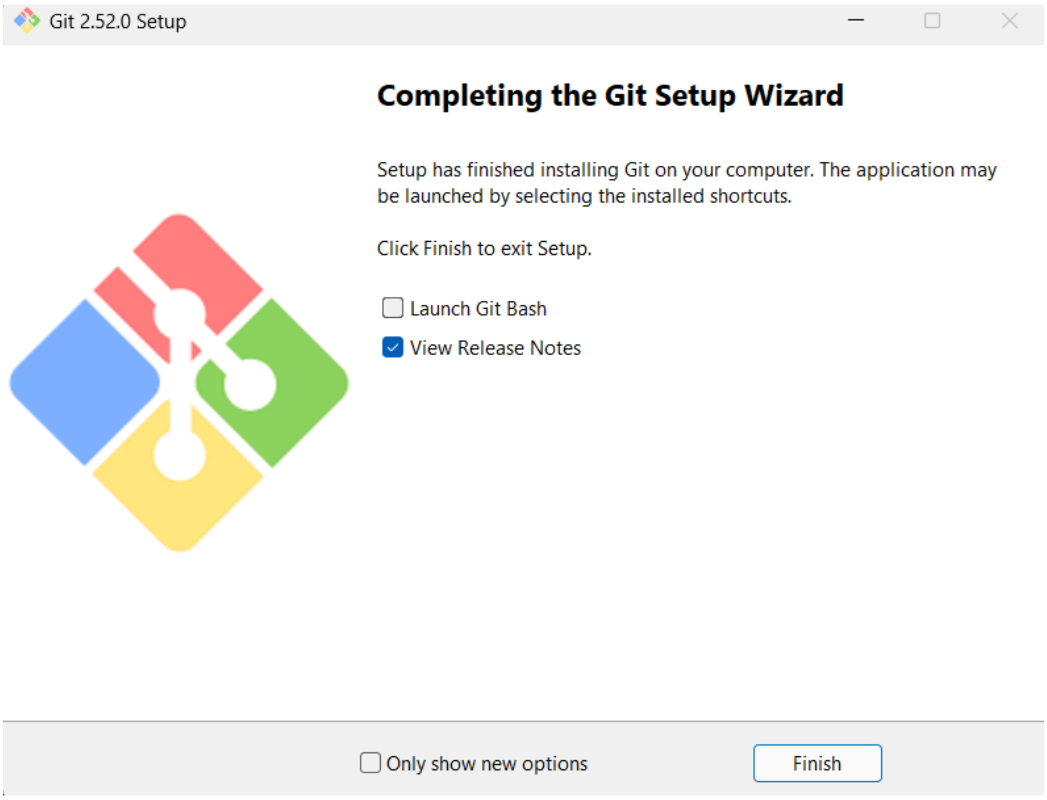
Please wait while Setup installs Git on your computer.

Extracting files...
C:\Program Files\Git\mingw64\bin\scalar.exe

https://gitforwindows.org/

Step 4: Finish Installation

- Click Finish



Git 2.52.0 Setup

Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

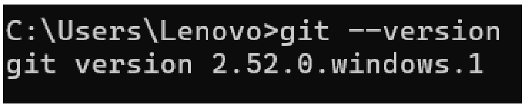
Click Finish to exit Setup.

□ Launch Git Bash

☑ View Release Notes

□ Only show new options

Verify Git Installation



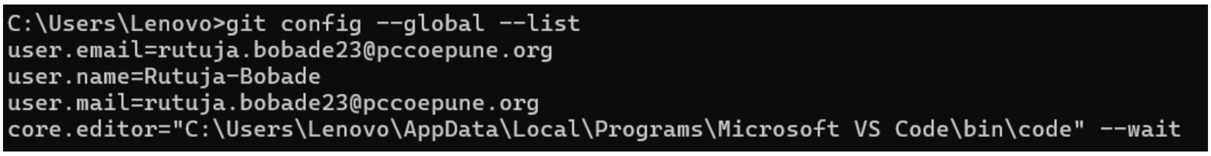
```
C:\Users\Lenovo>git --version
git version 2.52.0.windows.1
```

First-Time Git Configuration

Set your username:
git config --global user.name "Your Name"

Set your email ID:
git config --global user.email your_email@example.com

Verify configuration:
git config --global --list



```
C:\Users\Lenovo>git config --global --list
user.email=rutuja.bobade23@pccoepune.org
user.name=Rutuja-Bobade
user.mail=rutuja.bobade23@pccoepune.org
core.editor="C:\Users\Lenovo\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
```

1. Difference between Git and GitHub

Git	GitHub
Git is a version control software tool.	GitHub is a web-based service/platform
Installed on the local system	Hosted on the internet (cloud)
Used to track and manage versions of source code	Used to store and share Git repositories online
Works mainly through command-line interface	Provides a graphical user interface (GUI)
Can work offline	Requires internet connection
Manages local repositories	Hosts remote repositories for collaboration

2. Types of Version Control Systems

Version Control Systems are classified into three types:

i) Local Version Control Systems (Local VCS)

A Local Version Control System operates entirely on your personal machine without any connection to a remote repository. All changes and version history are stored in a local database on your computer.

In this setup, there is only a single user, with no collaboration or sharing of changes. Local VCS stores versions in a local database, not in a repository that others can clone.

Characteristics:

- No internet or server dependency.
- Useful for individual projects.
- Limited to single-user environments.

ii) Centralized Version Control System (CVCS)

In a Centralized Version Control System, all the files and their version history are stored in a single central server. Developers connect to this server to access or modify files.

The typical workflow is:

1. **Update/Checkout:** A developer pulls the latest version of the files from the central server.
2. **Make Changes:** A developer works on the files.

3. **Commit:** A developer saves ("commits") their changes directly back to the central server, making them immediately available to everyone else.

iii) Distributed Version Control System (CVCS)
Distributed version control systems contain multiple repositories. Each user has his or her own repository and working copy. Just committing your changes will not give others access to your changes. This is because a commit will reflect those changes in your local repository and you need to push them in order to make them visible to the central repository.

Similarly, When you update, you do not get others changes unless you have first pulled those changes into your repository.

The key difference is the two-step process for sharing changes:

1. **Commit:** You save your changes to your own local repository. At this point, the changes are only on your machine; no one else can see them.
2. **Push:** You upload ("push") your committed changes from your local repository to the central repository (e.g., GitHub)

3. What is .gitignore?

.gitignore is a special file used in Git to specify which files or folders should NOT be tracked by Git.

Purpose of .gitignore:

- Prevents unnecessary files from being committed
- Keeps repository clean and secure

Commonly ignored files:

- Temporary files
- Log files
- Compiled files
- Environment files

Example of .gitignore file:

```
node_modules/
.env
*.log
```

Conclusion
In this assignment, we studied the fundamentals of Git and Version Control Systems. We learned how to install Git on a Windows system, configure it with a username and email ID, and verify the installation. The assignment also helped in understanding the Git workflow, including the working directory, staging area, local repository, and remote repository.
Additionally, we explored the differences between Git and GitHub, various types of version control systems, and the importance of files like .gitignore. Overall, this assignment provided practical and theoretical knowledge of Git, which is essential for efficient code management, collaboration, and software development in real-world projects.