

Why Git and GitHub

Git is a distributed VCS that tracks changes to files over time, allowing you to revert mistakes, collaborate without overwriting work, and maintain a history of your project. It's efficient because it only stores changes, not full copies of files each time, saving space and enabling offline work.

GitHub is a cloud platform for hosting Git repositories, adding collaboration features like pull requests and issue tracking. Together, they make teamwork seamless—think of managing shared Cisco config scripts where multiple engineers contribute without chaos.

- + Collaboration: Multiple people edit files without conflicts (or resolve them easily).
- + History and Reversion: See who changed what and when; rollback errors quickly.
- + Branching: Work on features independently (e.g., test a new network script without breaking the main one).
- + Efficiency: Git is fast, lightweight, and works offline; GitHub provides backups and remote access.
- + Real-World Use: In DevNet, Git/GitHub manages Infrastructure as Code (IaC), preventing deployment disasters.

Lets do a shared "attendance" file:

1. I'll add my name, and you'll add yours, pushing changes to see version control in action.
2. If conflicts arise (e.g., two students edit simultaneously), Git will help resolve them (I hope so!!!).

Prerequisites:

- + A computer with internet access.
- + Windows/Linux OS. (Use your DevASC VM or your Host Machine)

Part 1

1. Create a GitHub Account (if not created 😊)
2. Go to <https://github.com> and click Sign up.
3. Enter your email, create a password, and choose a username (e.g., yourname-student).
4. Verify your email by clicking the link GitHub sends.
5. Once done, you'll have a free GitHub account.
6. Share this with your Lecturer via Vevox Survey

Give me some time: I will add you to our GitHub Repo



Part 2

Install Git on Your Machine

Git must be installed locally to interact with repositories. Download from [git-scm.com](https://git-scm.com/downloads).

In the command Prompt:

```
C:\Users\n.kaliyamurthy>git --version  
git version 2.47.0.windows.1
```

If you are not getting the version, Git is not installed. Then Install

For Windows Users:

1. Download the installer from <https://git-scm.com/download/win>
2. Run the .exe file.
3. Follow the wizard:
 - a. Accept defaults for most options.
 - b. Choose "Use Git from the Windows Command Prompt" (or Git Bash if you prefer a Unix-like terminal).
 - c. Select "Checkout as-is, commit Unix-style line endings" for compatibility.
 - d. Enable the experimental built-in file system monitor if prompted.
4. Finish the installation.
5. Open Command Prompt.
6. Verify:

```
C:\Users\n.kaliyamurthy>git --version  
git version 2.47.0.windows.1
```

For Linux Users:

1. Open your Terminal (Ctrl + Alt + T).
2. Update your package list: `sudo apt update`.
3. Install Git: `sudo apt install git`.
4. Enter your password if prompted.
5. Verify: Type `git --version`. You should see something like `git version 2.x.x`.

Part 3

Configure

In your terminal/Command Prompt:

1. `git config --global user.name "Your Full Name"`
2. `git config --global user.email "your.email@example.com"`
(Use the email tied to your GitHub account.)

```
C:\Users\n.kaliyamurthy>git --version
git version 2.47.0.windows.1

C:\Users\n.kaliyamurthy>git config --global user.name "Nitheesh"

C:\Users\n.kaliyamurthy>git config --global user.email "dummy.demo4class@gmail.com"
```

3. Instructor Sets Up the Repository and creates a File to work/ collaborate
4. Clone the Repository Locally (*Cloning creates a local copy of the repo on your machine*)
 - a. Open your terminal/Command Prompt (Windows: Use command prompt; Linux: Terminal).
 - b. Navigate to where you want the folder (e.g., create a projects folder)
 - c. Windows/Linux: `mkdir projects` then `cd projects`.
 - d. Clone the repo (replace with the actual URL from instructor):

```
C:\Users\n.kaliyamurthy>cd Downloads
C:\Users\n.kaliyamurthy\Downloads>mkdir week3
C:\Users\n.kaliyamurthy\Downloads>dir
Volume in drive C is OS
Volume Serial Number is EEA7-A74F

Directory of C:\Users\n.kaliyamurthy\Downloads
```

```
C:\Users\n.kaliyamurthy\Downloads>git clone https://github.com/UWTSO-SoAC-Cyber/ACCB5029-25
Cloning into 'ACCB5029-25'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 9.17 KiB | 1.31 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

```
C:\Users\n.kaliyamurthy\Downloads>dir /ad
Volume in drive C is OS
Volume Serial Number is EEA7-A74F

Directory of C:\Users\n.kaliyamurthy\Downloads

14/10/2025  06:32  <DIR>      .
14/10/2025  06:18  <DIR>      ..
14/10/2025  06:32  <DIR>      ACCB5029-25
```

Part 3: Troubleshooting

- **Error: Repository not found:** Ensure the URL is correct and you're a collaborator. Check with the instructor if you can't access it.
- **Authentication failed:** Use a Personal Access Token, not your GitHub password. Re-generate if needed.
- **Command not found:** Verify Git is installed (`git --version`). If not, install it.
- **Windows Path Issues:** Command Prompt may need `cd` with backslashes (e.g., `cd \projects`).
- **Username:** Your GitHub username (e.g., `yourusername`).
- **Password:** Use a **Personal Access Token** (not your regular password, as GitHub deprecated password auth for Git operations):
 - Go to GitHub > Profile (top-right) > Settings > Developer settings > Personal access tokens > Tokens (classic) > Generate new token.
 - Check scopes: `repo` (full control for private repos).
 - Copy the token and use it as your password when prompted.

Part 4

Edit the File, Commit, and Push Changes

1. Open the file locally:
 - a. Navigate to the folder: ``cd attendance``.
 - b. Open ``attendance.txt`` in a text editor:
 - i. Windows: Right-click > Open with Notepad (or use VS Code if installed).
 - ii. Linux: Use ``nano attendance.txt`` (or ``vim``, or a GUI editor like `gedit`).
 - c. Add your name on a new line: e.g., ``Student: Your Name``.
 - d. Save and close the file.
2. Back in terminal/Command Prompt, stage the change:

```

C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25>cd general
C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25\general>dir
Volume in drive C is OS
Volume Serial Number is EEA7-A74F

Directory of C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25\general

14/10/2025  06:32    <DIR>        .
14/10/2025  06:32    <DIR>        ..
14/10/2025  06:32                44 className
14/10/2025  06:32                19 info
                2 File(s)                63 bytes
                2 Dir(s)  272,411,852,800 bytes free

C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25\general>git add className.txt
fatal: pathspec 'className.txt' did not match any files

C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25\general>git add className

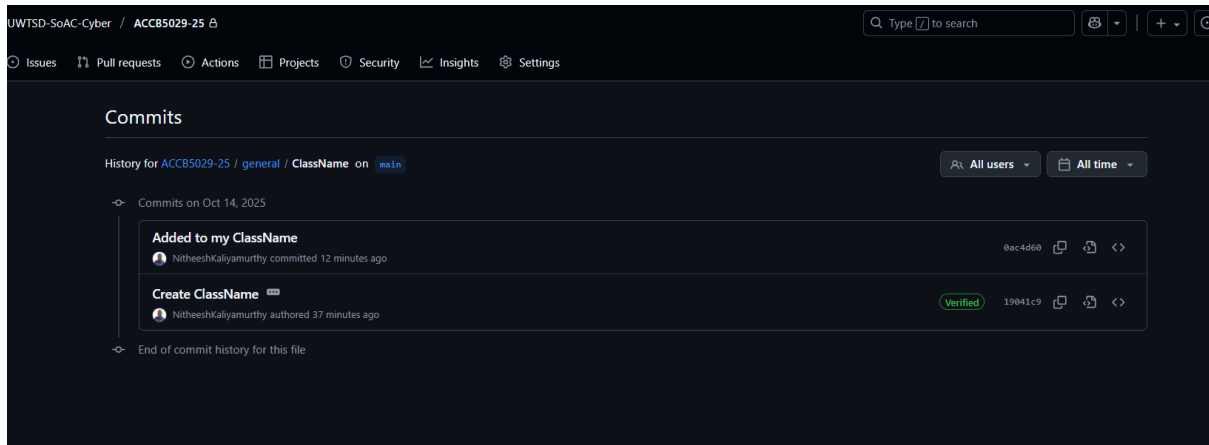
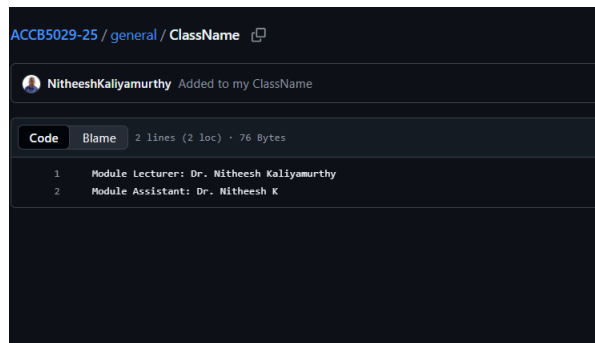
C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25\general>git commit -m "Added to my className"
[main 0ac4d60] Added to my className
 1 file changed, 1 insertion(+)

C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25\general>git pull origin main
From https://github.com/UWTSO-SoAC-Cyber/ACCB5029-25
 * branch      main      -> FETCH_HEAD
Already up to date.

C:\Users\n.kaliyamurthy\Downloads\ACCB5029-25\general>git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 397 bytes | 198.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/UWTSO-SoAC-Cyber/ACCB5029-25
 19041c9..0ac4d60  main -> main

```

- a. Commit with a message: `git commit -m "...."`
 - b. Pull latest changes (important to avoid conflicts if others pushed first): `git pull origin main`
 If conflicts arise (e.g., another student added their name simultaneously), Git will mark them in the file—edit to resolve, then ``git add`` and ``git commit`` again.
 - c. Push to GitHub: `git push origin main`
 If prompted, enter your GitHub username and password (or set up SSH keys for easier future access—search "GitHub SSH setup").
3. Observe Version Control in Action
- a. Go to the repo on GitHub in your browser.
 - b. Click on ``general/className``.
 - c. Click ****History**** (top-right) to see all commits.
 - d. Click a commit to view changes (diffs): See what lines were added.
 - e. If conflicts happened, discuss how Git flagged them and required manual resolution—preventing data loss.



Optional: Cleanup

1. To remove your local copy (optional):

``cd ..` then `rm -rf class-repo` (Linux)`

or

``rd /s /q class-repo`` (Windows Command Prompt).

2. Experiment: Create a branch (``git checkout -b my-branch``), edit, commit, and merge back (``git checkout main; git merge my-branch; git push``).

If issues arise (e.g., authentication errors), check ``git status`` for hints or ask the instructor.

This activity shows Git's efficiency in real collaboration: now you're ready for more advanced workflows!

Try using GitHub Desktop/ GitGUI apps and try out!!!