# 1. Setting Up and Basic Commands

This activity focuses on initializing a Git repository, staging and committing changes, and pushing them to a remote repository on GitHub.

Steps:

**1. Create a new repository on GitHub and copy the URL.**

2. Open VS Code and create a new folder.

3. Inside the folder, create a file (e.g., index.html) with content.

4. Open the terminal and run the following commands:

git init  
 git add .  
 git commit -m "Initial"  
 git remote add origin <repo-url>  
 git remote -v  
 git push -u origin main

Expected Output:

- The file is pushed successfully to GitHub and appears in the repository.

# 2. Creating and Managing Branches

This activity explains how to create new branches, switch between them, make changes, and merge them back into the main branch.

Steps:

1. Create and switch to a new branch:

git branch feature-branch  
 git checkout feature-branch  
 git branch -v

2. Make changes or add a file, then run:

git add .  
 git commit -m "Added feature file"

3. Switch back to main and merge:

git checkout main  
 git merge feature-branch  
 git push

Expected Output:

- The changes from the feature branch are now in the main branch and visible on GitHub.

# 3. Collaboration and Remote Repositories

This activity demonstrates how to work collaboratively by cloning repositories, creating branches, and merging changes using a merge message.

Steps:

1. Clone the repository:

git clone <repo-url>  
 cd <repo-folder>

2. Create and switch to a new feature branch:

git checkout -b feature-branch

3. Make changes and merge with a message:

git add .  
 git commit -m "Feature commit"  
 git checkout main  
 git merge feature-branch --no-ff -m "Merge feature-branch"

4. Use fetch and rebase to sync with the latest changes:

git fetch  
 git rebase origin/main

Expected Output:

- Clean merge history with no fast-forward and synchronized commits.

# 4. Git Tags and Releases

Tagging helps in creating stable versions of the code, and releases can be created on GitHub for these tags.

Steps:

1. Create a local tag and push:

git tag v1.0  
 git push origin v1.0

2. Go to GitHub → Releases → New Release.

3. Choose the tag v1.0, enter a title, and publish.

Expected Output:

- A release entry is created on GitHub and available for download.

# 5. Advanced Git Operations

This activity covers cherry-picking commits from one branch to another without merging the entire branch.

Steps:

1. Make changes and commit them:

git add .  
 git commit -m "Feature commit"

2. View recent commits:

git log --oneline

3. Switch to main and cherry-pick a commit:

git checkout main  
 git cherry-pick <commit-id>

Expected Output:

- Only the selected commit is added to the main branch.

# 6. Analyzing and Changing Git History

This activity allows us to view logs based on filters, show detailed commit info, and revert changes when needed.

Steps:

1. List commits by author and date:

git log --author="YourName" --after="2025-01-01" --before="2025-12-31"

2. Show specific commit details:

git show <commit-id>

3. Revert a specific commit:

git revert <commit-id>

4. Display last five commits:

git log -n 5

Expected Output:

- Displayed commit history, revert confirmation, and successful undo of changes.