Homework Exercises – Version Control (Git & GitHub)

Exercise 1: Install & Initialize

Set up Git on your local machine.

Commands used:

git --version

git init

Steps:

Install Git if not installed.

Verify installation using: `git --version`.

Create a new folder and open it in your terminal.

Initialize Git repository: `git init`.

Exercise 2: First Commit

Practice basic Git commands.

Commands used:

git status

git add hello.txt

git commit -m "Initial commit with hello.txt"

Steps:

Create a file `hello.txt` and add content inside.

Check file status: `git status`.

Add file to staging area: `git add hello.txt`.

Commit the file: `git commit -m "Initial commit with hello.txt"`.

Exercise 3: Using .gitignore

Ignore unnecessary files.

Commands used:

git status

Steps:

Create a folder called `temp/` and a file `notes.log`.

Create a `.gitignore` file and add:
`temp/
*.log`

Check if Git is ignoring them by running: `git status`.

Exercise 4: Branching and Merging

Understand branching workflow.

Commands used:

git checkout -b feature-readme

git add README.md

git commit -m "Add README file"

git checkout main

git merge feature-readme

Steps:

Create a new branch: `git checkout -b feature-readme`.

Add a new `README.md` file and commit it.

Switch back to `main`: `git checkout main`.

Merge branch into main: `git merge feature-readme`.

Resolve merge conflicts if they happen.

Exercise 5: Connect to GitHub

Push code to a remote repository.

Commands used:

git remote add origin https://github.com/your-username/your-repo.git

git push -u origin main

Steps:

Create a new repository on GitHub.

Connect local repo to remote:
`git remote add origin https://github.com/your-username/your-repo.git`

Push code to GitHub:
`git push -u origin main`.

Exercise 6: Pull Requests Simulation

Simulate team collaboration.

Steps:

Clone a shared repo.

Make a change locally and push it.

Create a Pull Request (PR) on GitHub.

Review and approve your partner's PR.

Exercise 7: Commit Message Practice

Practice writing meaningful commit messages.

Steps:

Write a commit message using this format:

[Type] Short description

- What changed?
- Why it was necessary?

Example commit message:

[Fix] Correct typo in README

- Fixed spelling of "installation" in README.md.
- Improves professionalism and clarity.

Bonus Challenge:

Create alias shortcuts for Git commands inside your `.bash_aliases` or `.zshrc` file.

Steps:

Example:
`alias gs="git status"
alias gc="git commit"
alias gp="git push"`