Assignment on Git

**Question 1: What is Git, and what problem does it solve in the context of software development?**

Git is a distributed version control system that tracks changes in source code during software development. It allows multiple developers to collaborate on a project simultaneously without overwriting each other’s work. Git solves the problem of managing codebase versions, enabling developers to work on different features, fix bugs, and experiment with new code branches while keeping track of changes, merging, and reverting if necessary.

**Question 2: Explain the difference between Git and GitHub.**

Git is the version control system itself, a tool for tracking changes in code. GitHub is a web-based platform that provides hosting for Git repositories. It adds collaboration features like issue tracking, pull requests, and code reviews, making it easier for developers to work together, share code, and manage projects.

**Question 3: Describe the basic workflow for using Git for version control.**

The basic Git workflow involves these steps:

1. Initialize a repository: Create a new repository or clone an existing one.
2. Make changes: Modify files within the repository.
3. Stage changes: Select which files to include in the next commit.
4. Commit changes: Record changes with a commit message explaining what was done.
5. Repeat steps 2-4: Continue making changes and committing as needed.
6. Push changes: Upload commits to a remote repository (like GitHub) to share with others.
7. Pull changes: Sync with remote changes to avoid conflicts.
8. Resolve conflicts (if any): Merge or resolve differences between local and remote changes.

**Question 4: What is a Git repository? How does it store and manage versions of files?**

A Git repository is a directory that contains all the files, folders, and version history of a project. It stores versions as commits, which are snapshots of the project at specific points in time. Commits are linked together, forming a history of changes. Git uses data structures to store files efficiently and allows branching and merging for parallel development.

**Question 5: What is a commit in Git? What information does a commit include?**

A commit in Git represents a specific version of the project. It includes:

* A unique identifier (hash) for the commit.
* The author’s name and email.
* A timestamp of when the commit was made.
* A commit message explaining the changes made.
* A pointer to the parent commit(s), forming a history.

**Question 6: Explain the concept of branching in Git. How can branching be useful in collaborative development?**

Branching in Git allows developers to work on separate lines of development within the same repository. Each branch represents a separate version of the codebase. Developers can create feature branches to work on new features or bug fixes without affecting the main code. Once the changes are tested and reviewed, they can be merged back into the main branch (often called “master” or “main”). This is useful for collaborative development as it allows multiple developers to work on different features simultaneously, keeping the main branch stable and avoiding conflicts until changes are ready to be integrated.