**Basics of Version Control Systems**

Version control systems (VCS) are tools that help manage changes to software code over time. They enable tracking modifications, reverting to previous versions, and collaboration among developers. Key concepts include:

* **Repository**: A central place where VCS stores data about your project (like Git repositories hosted on GitHub, GitLab, etc.).
* **Commit**: A snapshot of your repository at a given time. Each commit has a unique identifier and contains changes made since the last commit.
* **Branch**: A parallel version of the repository. Branches allow developers to work on features or fixes independently of the main codebase.
* **Merge**: Combining changes from one branch into another (e.g., merging a feature branch into the main branch).

**Core Git Commands and Workflows**

1. **Initializing a Repository**:
   * git init: Initializes a new Git repository in the current directory.
2. **Basic Workflow**:
   * git add <file>: Adds changes in <file> to the staging area.
   * git commit -m "message": Commits staged changes with a descriptive message.
3. **Checking Status and History**:
   * git status: Shows the current state of the working directory and staging area.
   * git log: Displays a history of commits.
4. **Working with Remotes**:
   * git remote add origin <remote-url>: Adds a remote repository (like GitHub).
   * git push -u origin <branch>: Pushes local changes to the remote repository.
   * git pull origin <branch>: Fetches and merges changes from the remote repository.

**Techniques for Branching, Merging, and Conflict Resolution**

1. **Branching**:
   * git branch: Lists all branches.
   * git branch <branch-name>: Creates a new branch.
   * git checkout <branch-name>: Switches to another branch.
2. **Merging**:
   * git merge <branch>: Combines changes from <branch> into the current branch.
3. **Conflict Resolution**:
   * Conflicts occur when Git cannot automatically merge changes from different branches. You resolve conflicts by:
     + Editing the conflicting files to remove markers and integrate the changes manually.
     + Using git mergetool to launch a graphical tool to resolve conflicts.