Introduction to Git and GitHub for Version Control

1. Setting Up Git

Installing Git:

- **Windows**: Download from git-scm.com and install.
- **Mac**: Use Homebrew:

  ```
  brew install git
  ```

- **Linux**: Use package manager:
  - **Debian/Ubuntu**:

    ```
    sudo apt install git
    ```

  - **Fedora**:

    ```
    sudo dnf install git
    ```

  - **Arch**:

    ```
    sudo pacman -S git
    ```

Configuring Git:

After installation, configure Git with your username and email:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

To verify the configuration:

```
git config --list
```

2. Understanding Version Control

Version Control System (VCS):

Tracks changes to files over time.

Types:

- **Local VCS**: Simple tracking on a local machine.
- **Centralized VCS**: Single repository accessible to all developers (e.g., SVN).
- **Distributed VCS**: Every developer has a full copy of the repository (e.g., Git).

3. Basic Git Commands

Initializing a Repository

```
git init
```

Creates a new Git repository in the current directory.

Cloning a Repository

```
git clone <repository-url>
```

Creates a copy of an existing repository from GitHub.

Checking Repository Status

```
git status
```

Shows the current state of the working directory and staged files.

Adding Changes to Staging Area

```
git add <filename>
```

Stages a file for commit.

To add all files:

```
git add .
```

Committing Changes

```
git commit -m "Commit message"
```

Saves changes to the repository with a descriptive message.

Viewing Commit History

```
git log
```

Displays a history of commits.

4. Working with Branches

Creating a New Branch

```
git branch <branch-name>
```

Switching to a Branch

```
git checkout <branch-name>
```

(Or use the newer command)

```
git switch <branch-name>
```

Merging Branches

```
git merge <branch-name>
```

Merges a specified branch into the current branch.

5. Pushing and Pulling from GitHub

Connecting Local Repository to GitHub

```
git remote add origin <repository-url>
```

Pushing Changes to GitHub

```
git push origin <branch-name>
```

Sends local commits to the remote repository.

Pulling Changes from GitHub

```
git pull origin <branch-name>
```

Updates local repository with changes from GitHub.

6. Resolving Merge Conflicts

Occurs when two branches modify the same file. Git marks conflict sections, and manual resolution is needed.

After resolving conflicts:

```
 git add .
git commit -m "Resolved merge conflicts"
```

7. Additional Useful Commands

- **Check Differences**:

  ```
  git diff
  ```

- **Remove a File from Staging**:

  ```
  git reset <file>
  ```

- **Undo Last Commit**:

  ```
  git reset --soft HEAD~1
  ```

- **Stash Changes**:

  ```
  git stash
  ```

- **List Remote Repositories**:

  ```
  git remote -v
  ```

8. Best Practices for Git and GitHub

- Use meaningful commit messages.
- Keep branches short-lived and merge frequently.
- Pull latest changes before making new commits.
- Use `.gitignore` to exclude unnecessary files.
- Protect the main or master branch with branch policies.