ABES Engineering College

Department of Computer Science and Engineering

Git and GitHub Introduction

Step-by-Step Guide for 2nd Year Students (Windows)

1. What are Git and GitHub?

- **Git**: Local version control system to track changes in code.
- **GitHub**: Cloud platform to host Git repositories and collaborate.

2. Installing Git on Windows

1. Download Git for Windows from `https://git-scm.com`.
2. Run installer and select options:
   - Editor: Visual Studio Code recommended.
   - PATH: *Git from command line and also from 3rd-party software.*
   - OpenSSH: Use bundled OpenSSH.
   - HTTPS backend: Use OpenSSL library.
   - Line endings: Checkout Windows-style, commit Unix-style (CRLF → LF).
   - Default branch name: `main`.
   - Credential helper: Git Credential Manager.
3. Verify installation with:

```
git --version
```

3. Installing VS Code (Optional)

- Download from `https://code.visualstudio.com`.
- Install GitLens and GitHub Pull Requests extensions.

4. Creating a GitHub Account

- Sign up at `https://github.com`.
- Choose a username carefully.
- Enable two-factor authentication for security.

5. First-Time Git Configuration

```
git config --global user.name  "Your Name"
git config --global user.email "you@example.com"
git config --global init.defaultBranch main
git config --global core.autocrlf true
git config --global core.editor "code --wait"
git config --global alias.lg "log --oneline --graph --
   decorate --all"
git config --global --list
```

6. Authentication Options

HTTPS (Easy)

- Git Credential Manager stores login info securely.

SSH Keys (Advanced)

```
ssh-keygen -t ed25519 -C "you@example.com"
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
cat ~/.ssh/id_ed25519.pub
ssh -T git@github.com
```

7. First Project (Local to GitHub)

```
mkdir hello-git && cd hello-git
git init
echo "# Hello Git" > README.md
git add .
git commit -m "Initialize project"
git remote add origin https://github.com/<user>/hello-git.git
git push -u origin main
```

8. Cloning an Existing Repo

```
git clone https://github.com/<org>/<repo>.git
```

9. Daily Workflow

```
git status
git add .
git commit -m "Message"
git pull --rebase
git push
```

10. Branching and Pull Requests

```
git switch -c feature/login-ui
git add .
git commit -m "Add login UI"
git push -u origin feature/login-ui
```

Open a pull request on GitHub, get reviews, and merge.

11. Handling Merge Conflicts

1. Pull changes, conflict markers appear.
2. Edit files to resolve conflicts.
3. Stage and commit resolved files.

12. Undoing Mistakes

```
git restore --staged <file>
git restore <file>
git revert <commit-hash>
git reset --soft HEAD~1
```

13. Useful .gitignore Entries

```
# Windows & VS Code
Thumbs.db
.vscode/

# Node.js
```

```
node_modules/
dist/
.env

# Java
*.class
target/
build/
.gradle/
```

14. Cheatsheet

```
git init
git clone <url>
git status
git add .
git commit -m "Message"
git log
git switch -c feature/x
git push -u origin main
git pull --rebase
```

15. Common Errors

- **fatal: not a git repository** → Run inside repo folder.
- **Permission denied (publickey)** → Configure SSH keys or use HTTPS.
- **Auth prompts every time** → Use Credential Manager.
- **Updates were rejected** → Run `git pull --rebase`.

16. Practice Task for Students

1. Install Git and set config.
2. Create a project, commit, push to GitHub.
3. Create a feature branch, make changes, open a Pull Request, merge.