**Week - 8**

Exercise 1: Git Configuration & Notepad++ Integration

Checking Git Installation:







git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

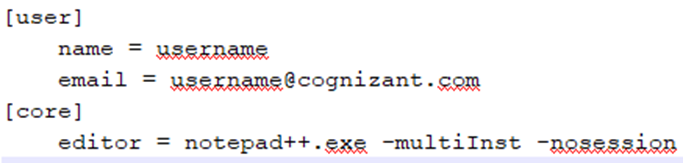
3. Verify Configuration:

git config --list

4. Check if Notepad++ Works:

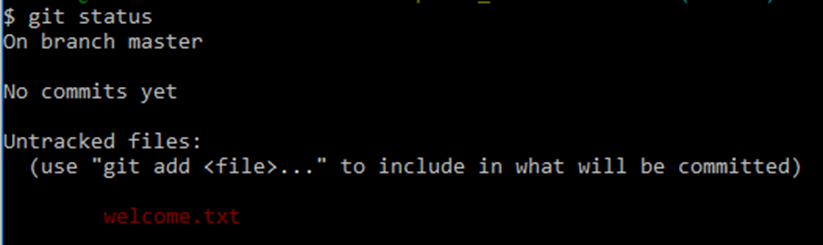
notepad++

5. Add Notepad++ to PATH (via Environment Variables in Windows)



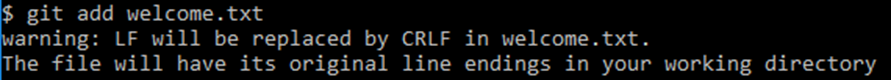
6. Create Alias for Notepad++:

alias np="notepad++"



7. Set as Default Git Editor:

git config --global core.editor "notepad++ -multiInst -nosession"



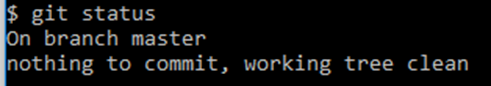
8. Verify Default Editor:

git config --global -e

9. Create Repository:

mkdir GitDemo && cd GitDemo

git init



10. Create File:

echo "Welcome to Git!" > welcome.txt

git add welcome.txt

git commit

11. Add Remote and Push:

git remote add origin <remote-url>

git pull origin master

git push origin master

**Exercise 2: Git Ignore**

1. Create a .log file and log folder:

echo "Log entry" > error.log

mkdir log

echo "Another log" > log/test.log

2. Create/Edit .gitignore:

echo "\*.log" > .gitignore

echo "log/" >> .gitignore

3. Stage & Commit:

git add .gitignore

git commit -m "Added .gitignore to exclude log files and folder"

4. Verify Status:

git status

**Exercise 3: Branching & Merging**

1. Create Branch:

git branch GitNewBranch

2. List Branches:

git branch -a

3. Switch to Branch:

git checkout GitNewBranch

4. Add File and Commit:

echo "New content" > branchfile.txt

git add branchfile.txt

git commit -m "Added file in branch"

5. Switch to Master:

git checkout master

6. View Differences:

git diff master GitNewBranch

7. Merge Branch:

git merge GitNewBranch

8. View Log:

git log --oneline --graph --decorate

9. Delete Branch:

git branch -d GitNewBranch

**Exercise 4: Conflict Resolution**

1. Verify Master Clean State:

git status

2. Create Branch and Add File:

git checkout -b GitWork

echo "Branch content" > hello.xml

git add hello.xml

git commit -m "Added hello.xml in branch"

3. Switch to Master and Add Conflicting File:

git checkout master

echo "Master content" > hello.xml

git add hello.xml

git commit -m "Added hello.xml in master"

4. Merge and Resolve Conflict:

git merge GitWork

git add hello.xml

git commit -m "Resolved merge conflict for hello.xml"

5. Add Backup File to .gitignore:

echo "\*.bak" >> .gitignore

git add .gitignore

git commit -m "Ignored backup files"

6. Delete Branch:

git branch -d GitWork

git log --oneline --graph --decorate

**Exercise 5: Clean Up and Push to Remote Git**

1. Verify Master Clean State:

git status

2. List All Branches:

git branch -a

3. Pull Latest Changes from Remote:

git pull origin master

4. Push Pending Changes:

git push origin master

5. Verify Changes in Remote:

# Check GitLab/GitHub to confirm updates are visible