

Assignment 1: Git Basics

Familiarize with git init, git status, git add, git commit, git push, git pull, configuring Git, setting Notepad++ as default editor, and adding a file to the repository.

Step 1: Git Configuration

```
git --version
```

Output:

```
Last login: Sun Aug 10 19:15:22 on ttys000
(base) apple@sais-macbook-pro ~ % git --version
git version 2.39.5 (Apple Git-154)
```

```
git config --global user.name "Sai Kushyanth"
```

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

```
git config --list
```

Output:

```
(base) apple@sais-macbook-pro ~ % git config --global user.name "Sai Kushyanth"
(base) apple@sais-macbook-pro ~ % git config --global user.email
"saikushyanth@gmail.com"
(base) apple@sais-macbook-pro ~ % git config --list
credential.helper=osxkeychain
init.defaultbranch=main
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
user.name=Sai Kushyanth
user.email=saikushyanth@gmail.com
```

Step 2: Create GitDemo Repository

```
mkdir GitDemo && cd GitDemo
```

```
git init
```

Output:

```
(base) apple@sais-macbook-pro ~ % mkdir GitDemo && cd GitDemo
(base) apple@sais-macbook-pro GitDemo % git init
Initialized empty Git repository in /Users/apple/GitDemo/.git/
```

```
echo "Welcome to Git" > welcome.txt
```

```
cat welcome.txt
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % echo "Welcome to Git" > welcome.txt
```

```
(base) apple@sais-macbook-pro GitDemo % cat welcome.txt
```

```
Welcome to Git
```

```
git status
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git status
```

```
On branch main
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

welcome.txt

nothing added to commit but untracked files present (use "git add" to track)

```
git add welcome.txt
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git add welcome.txt
(base) apple@sais-macbook-pro GitDemo %
```

```
git commit -m 'Add welcome.txt'
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git commit -m 'Add welcome.txt'
[main (root-commit) c2aa3a0] Add welcome.txt
 1 file changed, 1 insertion(+)
 create mode 100644 welcome.txt
(base) apple@sais-macbook-pro GitDemo %
```

```
git remote add origin <https://github.com/sai-kushyanth/Cognizant-Nurtue-Jse-4.0.git>
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git remote
add origin https://github.com/sai-kushyanth/Cognizant-
Nurtue-Jse-4.0.git
(base) apple@sais-macbook-pro GitDemo %
```

```
git pull origin main
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git pull origin main
remote: Enumerating objects: 736, done.
remote: Counting objects: 100% (167/167), done.
remote: Compressing objects: 100% (120/120), done.
Receiving objects: 61% (454/736), 50.54 MiB | 1.06 MiB/s
```

Assignment 2: Git Ignore

Learn to ignore files and folders using .gitignore.

```
touch error.log
mkdir logs
echo "*.log" > .gitignore
echo "logs/" >> .gitignore
git status
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % touch error.log
(base) apple@sais-macbook-pro GitDemo % mkdir logs

mkdir: logs: File exists
(base) apple@sais-macbook-pro GitDemo % echo "*.log" > .gitignore
(base) apple@sais-macbook-pro GitDemo % echo "logs/"
>> .gitignore
(base) apple@sais-macbook-pro GitDemo % git status
```

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)

```
.gitignore
logs
mkdir
```

nothing added to commit but untracked files present (use "git add" to track)

```
(base) apple@sais-macbook-pro GitDemo %
```

Assignment 3: Branching & Merging

Learn to create branches, make changes, and merge.

```
git branch GitNewBranch
```

```
git branch
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git branch GitNewBranch
(base) apple@sais-macbook-pro GitDemo % git branch
```

```
GitNewBranch
```

```
* main
```

```
(base) apple@sais-macbook-pro GitDemo %
```

```
git checkout GitNewBranch
```

```
echo "Branch file" > branch.txt
```

```
git add branch.txt
```

```
git commit -m 'Add branch.txt'
```

```
git checkout master
```

```
git diff master GitNewBranch
```

```
git merge GitNewBranch
```

```
git log --oneline --graph --decorate
```

```
git branch -d GitNewBranch
```

Output:

```
(base) apple@sais-macbook-pro GitDemo
% git branch GitNewBranch

(base) apple@sais-macbook-pro GitDemo % git branch

    GitNewBranch
* main
(base) apple@sais-macbook-pro GitDemo % git checkout GitNewBranch

Switched to branch 'GitNewBranch'
(base) apple@sais-macbook-pro GitDemo % echo "Branch file" >
branch.txt
(base) apple@sais-macbook-pro GitDemo % git add branch.txt
(base) apple@sais-macbook-pro GitDemo % git commit -m 'Add
branch.txt'
[GitNewBranch 02a596a] Add branch.txt
 1 file changed, 1 insertion(+)
 create mode 100644 branch.txt
(base) apple@sais-macbook-pro GitDemo % git checkout main
Switched to branch 'main'
(base) apple@sais-macbook-pro GitDemo % git diff main
GitNewBranch
diff --git a/branch.txt b/branch.txt
new file mode 100644
index 0000000..dd0358d
--- /dev/null
+++ b/branch.txt
@@ -0,0 +1 @@
+Branch file
(base) apple@sais-macbook-pro GitDemo % git merge GitNewBranch

Updating c2aa3a0..02a596a
Fast-forward
 branch.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 branch.txt
(base) apple@sais-macbook-pro GitDemo % git log --oneline --graph
--decorate
* 02a596a (HEAD -> main, GitNewBranch) Add branch.txt
* c2aa3a0 Add welcome.txt
(base) apple@sais-macbook-pro GitDemo % git branch -d
GitNewBranch
Deleted branch GitNewBranch (was 02a596a).
(base) apple@sais-macbook-pro GitDemo %
```

Assignment 4: Conflict Resolution

Resolve merge conflicts.

```
git checkout -b GitWork
echo "Branch hello" > hello.xml
git add hello.xml && git commit -m 'Add hello.xml in branch'
git checkout master
echo "Master hello" > hello.xml
```

```
git add hello.xml && git commit -m 'Add hello.xml in master'
```

```
git merge GitWork
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git checkout -b GitWork  
Switched to a new branch 'GitWork'
```

```
(base) apple@sais-macbook-pro GitDemo %
```

```
(base) apple@sais-macbook-pro GitDemo % echo "Branch hello" > hello.xml
```

```
(base) apple@sais-macbook-pro GitDemo % git add hello.xml && git commit -m 'Add  
hello.xml in branch'
```

```
[GitWork 60aa389] Add hello.xml in branch
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 hello.xml
```

```
(base) apple@sais-macbook-pro GitDemo % git checkout main
```

```
Switched to branch 'main'
```

```
(base) apple@sais-macbook-pro GitDemo % echo "Main hello" > hello.xml
```

```
(base) apple@sais-macbook-pro GitDemo % git add hello.xml && git commit -m 'Add  
hello.xml in main'
```

```
[main 8086339] Add hello.xml in main
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 hello.xml
```

```
(base) apple@sais-macbook-pro GitDemo % git merge GitWork
```

Auto-merging hello.xml

CONFLICT (add/add): Merge conflict in hello.xml

Automatic merge failed; fix conflicts and then commit the result.

```
(base) apple@sais-macbook-pro GitDemo %
```

```
git mergetool
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git mergetool
```

This message is displayed because 'merge.tool' is not configured.

See 'git mergetool --tool-help' or 'git help config' for more details.

'git mergetool' will now attempt to use one of the following tools:

opendiff tortoisemerge emerge vimdiff nvimdiff

Merging:

hello.xml

Normal merge conflict for 'hello.xml':

{local}: created file

{remote}: created file

Hit return to start merge resolution tool (opendiff):

```
git add hello.xml && git commit -m 'Resolved conflict'
```

```
echo "*.bak" >> .gitignore
```

```
git add .gitignore && git commit -m 'Ignore backup files'
```

```
git branch -d GitWork
```

Output:

Normal merge conflict for 'hello.xml':

{local}: created file

{remote}: created file

Hit return to start merge resolution tool (opendiff): git add hello.xml && git commit -m
'Resolved conflict'

xcode-select: error: tool 'opendiff' requires Xcode, but active developer directory '/Library/
Developer/CommandLineTools' is a command line tools instance

hello.xml seems unchanged.

Was the merge successful [y/n]? y

```
(base) apple@sais-macbook-pro GitDemo % echo "*.bak" >> .gitignore
```

```
(base) apple@sais-macbook-pro GitDemo % git add .gitignore && git commit -m 'Ignore backup files'
```

```
[main daedb12] Ignore backup files
```

```
(base) apple@sais-macbook-pro GitDemo % git branch -d GitWork
```

```
Deleted branch GitWork (was 60aa389).
```

```
(base) apple@sais-macbook-pro GitDemo %
```

Assignment 5: Cleanup & Push

Clean up local repo and push changes to remote.

```
git status
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git status
```

```
On branch main
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
hello.xml.orig
```

```
logs
```

```
mkdir
```

nothing added to commit but untracked files present (use "git add" to track)

```
(base) apple@sais-macbook-pro GitDemo %
```

```
git branch
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git branch
```

```
* main
```

```
git pull origin master
```

Output:

```
(base) apple@sais-macbook-pro GitDemo % git pull origin main
```

```
From https://github.com/sai-kushyanth/Cognizant-Nurtue-Jse-4.0
```

```
* branch          main          -> FETCH_HEAD
```

```
git push origin master
```

output:

```
(base) apple@sais-macbook-pro GitDemo % git push origin main
```

```
To https://github.com/sai-kushyanth/Cognizant-Nurtue-Jse-4.0.git  
main -> main
```