Name:Duggirala Bhagya Vivekini

Superset ID: 5673888

Cognizant Digital-Nurture-4.0-JavaFSE- Handson WEEK-8

1.Git-HOL Hands-On Solution

Setup Git Configuration:

git --version

git config --global user.name "Duggirala Bhagya Vivekini"

git config --global user.email "2200040307ece@gmail.com"

git config –list  
  
OUTPUT:

$ git --version

git version 2.42.0.windows.1

$ git config --global user.name "Duggirala Bhagya Vivekini"

$ git config --global user.email "2200040307ece@gmail.com"

$ git config --list

user.name=Duggirala Bhagya Vivekini

user.email=2200040307ece@gmail.com

core.editor='C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin

Integrate Notepad++ as Default Git Editor

notepad++

C:\Program Files\Notepad++\notepad++.exe

notepad++

alias np='"/c/Program Files/Notepad++/notepad++.exe"'

git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"  
git config --global -e

OUTPUT:

$ notepad++

$ alias np='"/c/Program Files/Notepad++/notepad++.exe"'

$ np

$ git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"

$ git config --global -e

Create Local Repository and Add File

mkdir GitDemo

cd GitDemo

git init

ls -a

echo "Welcome to Git Hands-On Lab!" > welcome.txt

ls

cat welcome.txt

git status

git add welcome.txt

git commit

git status

OUTPUT:

$ mkdir GitDemo

$ cd GitDemo

$ git init

Initialized empty Git repository in C:/Users/YourUsername/GitDemo/.git/

$ ls -a

. .. .git

$ echo "Welcome to Git Hands-On Lab!" > welcome.txt

$ ls

welcome.txt

$ cat welcome.txt

Welcome to Git Hands-On Lab!

$ git status

On branch master

No commits yet

Untracked files:

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

welcome.txt

$ git add welcome.txt

$ git commit

$ git status

On branch master

nothing to commit, working tree clean

Connect to Remote Repository

git remote add origin https://gitlab.com/yourusername/GitDemo.git

git pull origin master

git push origin master

OUTPUT:

$ git remote add origin https://gitlab.com/yourusername/GitDemo.git

$ git pull origin master

From https://gitlab.com/yourusername/GitDemo

\* branch master -> FETCH\_HEAD

Already up to date.

$ git push origin master

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 215 bytes | 215.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://gitlab.com/yourusername/GitDemo.git

\* [new branch] master -> master

2.Git-HOL Hands-On – .gitignore

Commands:

mkdir GitIgnoreDemo

cd GitIgnoreDemo

git init

echo "This is a log file" > error.log

mkdir log

echo "Log folder test" > log/info.txt

touch .gitignore

echo "\*.log" >> .gitignore

echo "log/" >> .gitignore

git status

git add .

git status

git commit -m "Initial commit ignoring .log files and log folder"

git status

Output:

$ mkdir GitIgnoreDemo

$ cd GitIgnoreDemo

$ git init

Initialized empty Git repository in C:/Users/YourName/GitIgnoreDemo/.git/

$ echo "This is a log file" > error.log

$ mkdir log

$ echo "Log folder test" > log/info.txt

$ touch .gitignore

$ echo "\*.log" >> .gitignore

$ echo "log/" >> .gitignore

$ git status

On branch master

No commits yet

Untracked files:

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

.gitignore

$ git add .

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: .gitignore

Untracked files:

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

error.log

log/

$ git commit -m "Initial commit ignoring .log files and log folder"

[master (root-commit) 3a6eab9] Initial commit ignoring .log files and log folder

1 file changed, 2 insertions(+)

create mode 100644 .gitignore

$ git status

On branch master

Untracked files:

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

error.log

log/

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

3.Git-HOL – Branching and Merging:

mkdir GitBranchDemo

cd GitBranchDemo

git init

git checkout -b GitNewBranch

git branch -a

echo "This is a test file in new branch" > branchfile.txt

git add branchfile.txt

git commit -m "Added branchfile.txt in GitNewBranch"

git status

git checkout master

git diff master GitNewBranch

git difftool master GitNewBranch

git merge GitNewBranch

git log --oneline --graph --decorate

git branch -d GitNewBranch

git status

Output:

$ mkdir GitBranchDemo

$ cd GitBranchDemo

$ git init

Initialized empty Git repository in C:/Users/YourName/GitBranchDemo/.git/

$ git checkout -b GitNewBranch

Switched to a new branch 'GitNewBranch'

$ git branch -a

\* GitNewBranch

master

$ echo "This is a test file in new branch" > branchfile.txt

$ git add branchfile.txt

$ git commit -m "Added branchfile.txt in GitNewBranch"

[GitNewBranch 7f3c2ab] Added branchfile.txt in GitNewBranch

1 file changed, 1 insertion(+)

create mode 100644 branchfile.txt

$ git status

On branch GitNewBranch

nothing to commit, working tree clean

$ git checkout master

Switched to branch 'master'

$ git diff master GitNewBranch

diff --git a/branchfile.txt b/branchfile.txt

new file mode 100644

index 0000000..32a3fa9

--- /dev/null

+++ b/branchfile.txt

@@ -0,0 +1 @@

+This is a test file in new branch

$ git difftool master GitNewBranch

# (P4Merge opens to show visual diff between master and GitNewBranch)

$ git merge GitNewBranch

Updating 1c7a95e..7f3c2ab

Fast-forward

branchfile.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 branchfile.txt

$ git log --oneline --graph --decorate

\* 7f3c2ab (HEAD -> master, GitNewBranch) Added branchfile.txt in GitNewBranch

\* 1c7a95e Initial commit

$ git branch -d GitNewBranch

Deleted branch GitNewBranch (was 7f3c2ab).

$ git status

On branch master

nothing to commit, working tree clean

4. Git-HOL – Merge Conflict Resolution:  
mkdir GitConflictDemo

cd GitConflictDemo

git init

git checkout -b GitWork

echo "<message>Hello from GitWork branch</message>" > hello.xml

git status

git add hello.xml

git commit -m "GitWork: Added hello.xml"

git checkout master

echo "<note>Hello from Master branch</note>" > hello.xml

git add hello.xml

git commit -m "Master: Added hello.xml with different content"

git log --oneline --graph --decorate --all

git diff master GitWork

git difftool master GitWork

git merge GitWork

# (Manually resolve conflict in hello.xml)

git status

git add hello.xml

git commit -m "Resolved conflict in hello.xml"

echo "\*.\*~" >> .gitignore

git add .gitignore

git commit -m "Added backup files to .gitignore"

git branch

git branch -d GitWork

git log --oneline --graph –decorate

Output:

$ mkdir GitConflictDemo

$ cd GitConflictDemo

$ git init

Initialized empty Git repository in C:/Users/YourName/GitConflictDemo/.git/

$ git checkout -b GitWork

Switched to a new branch 'GitWork'

$ echo "<message>Hello from GitWork branch</message>" > hello.xml

$ git status

On branch GitWork

Untracked files:

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

hello.xml

$ git add hello.xml

$ git commit -m "GitWork: Added hello.xml"

[GitWork 7a4b9c3] GitWork: Added hello.xml

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

$ git checkout master

Switched to branch 'master'

$ echo "<note>Hello from Master branch</note>" > hello.xml

$ git add hello.xml

$ git commit -m "Master: Added hello.xml with different content"

[master b32acb1] Master: Added hello.xml with different content

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

$ git log --oneline --graph --decorate --all

\* b32acb1 (HEAD -> master) Master: Added hello.xml with different content

|

| \* 7a4b9c3 (GitWork) GitWork: Added hello.xml

|/

$ git diff master GitWork

diff --git a/hello.xml b/hello.xml

index e69de29..f2d1c5c 100644

--- a/hello.xml

+++ b/hello.xml

@@ -1 +1 @@

-<note>Hello from Master branch</note>

+<message>Hello from GitWork branch</message>

$ git difftool master GitWork

# (P4Merge opens showing file differences)

$ git merge GitWork

Auto-merging hello.xml

CONFLICT (content): Merge conflict in hello.xml

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

$ git status

On branch master

You have unmerged paths.

(fix conflicts and run "git commit")

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: hello.xml

$ git add hello.xml

$ git commit -m "Resolved conflict in hello.xml"

[master 91fdc45] Resolved conflict in hello.xml

$ echo "\*.\*~" >> .gitignore

$ git add .gitignore

$ git commit -m "Added backup files to .gitignore"

[master b87d341] Added backup files to .gitignore

$ git branch

\* master

$ git branch -d GitWork

Deleted branch GitWork (was 7a4b9c3).

$ git log --oneline --graph --decorate

\* b87d341 (HEAD -> master) Added backup files to .gitignore

\* 91fdc45 Resolved conflict in hello.xml

\* b32acb1 Master: Added hello.xml with different content

\* 7a4b9c3 GitWork: Added hello.xml

5.Git-HOL – Cleanup and Push to Remote Git:

cd GitConflictDemo

git status

git branch

git pull origin master

git push origin master

Output:

$ cd GitConflictDemo

$ git status

On branch master

nothing to commit, working tree clean

$ git branch

\* master

$ git pull origin master

From https://gitlab.com/yourusername/GitConflictDemo

\* branch master -> FETCH\_HEAD

Already up to date.

$ git push origin master

Enumerating objects: 12, done.

Counting objects: 100% (12/12), done.

Delta compression using up to 4 threads

Compressing objects: 100% (6/6), done.

Writing objects: 100% (12/12), 1.02 KiB | 1.02 MiB/s, done.

Total 12 (delta 2), reused 0 (delta 0)

To https://gitlab.com/yourusername/GitConflictDemo.git

\* [new branch] master -> master