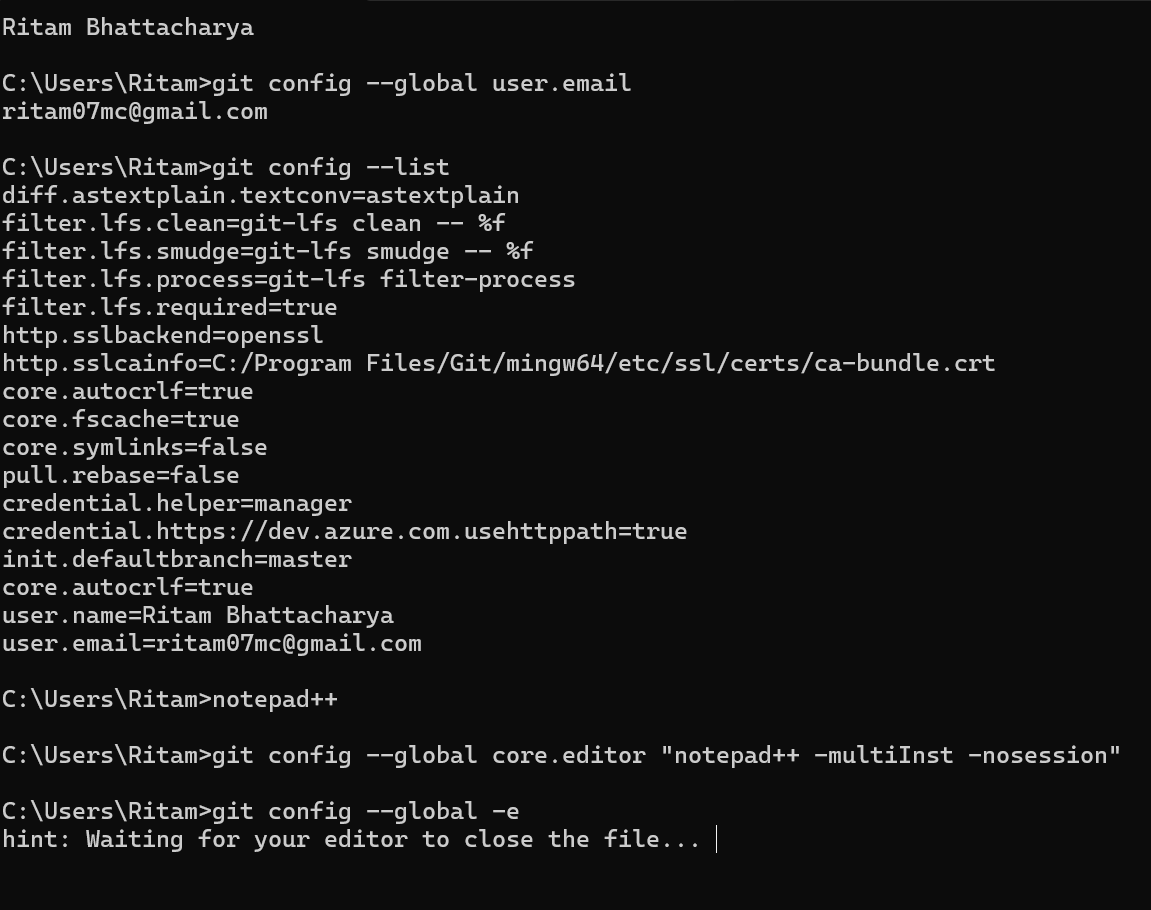
**Git HOL - 1**

## Part 1: Git Configuration and Notepad++ Integration

Commands Executed:

* git config --global user.email ritam07mc@gmail.com
* git config --list
* notepad++
* git config --global core.editor "notepad++ -multiInst -nosession"
* git config --global -e

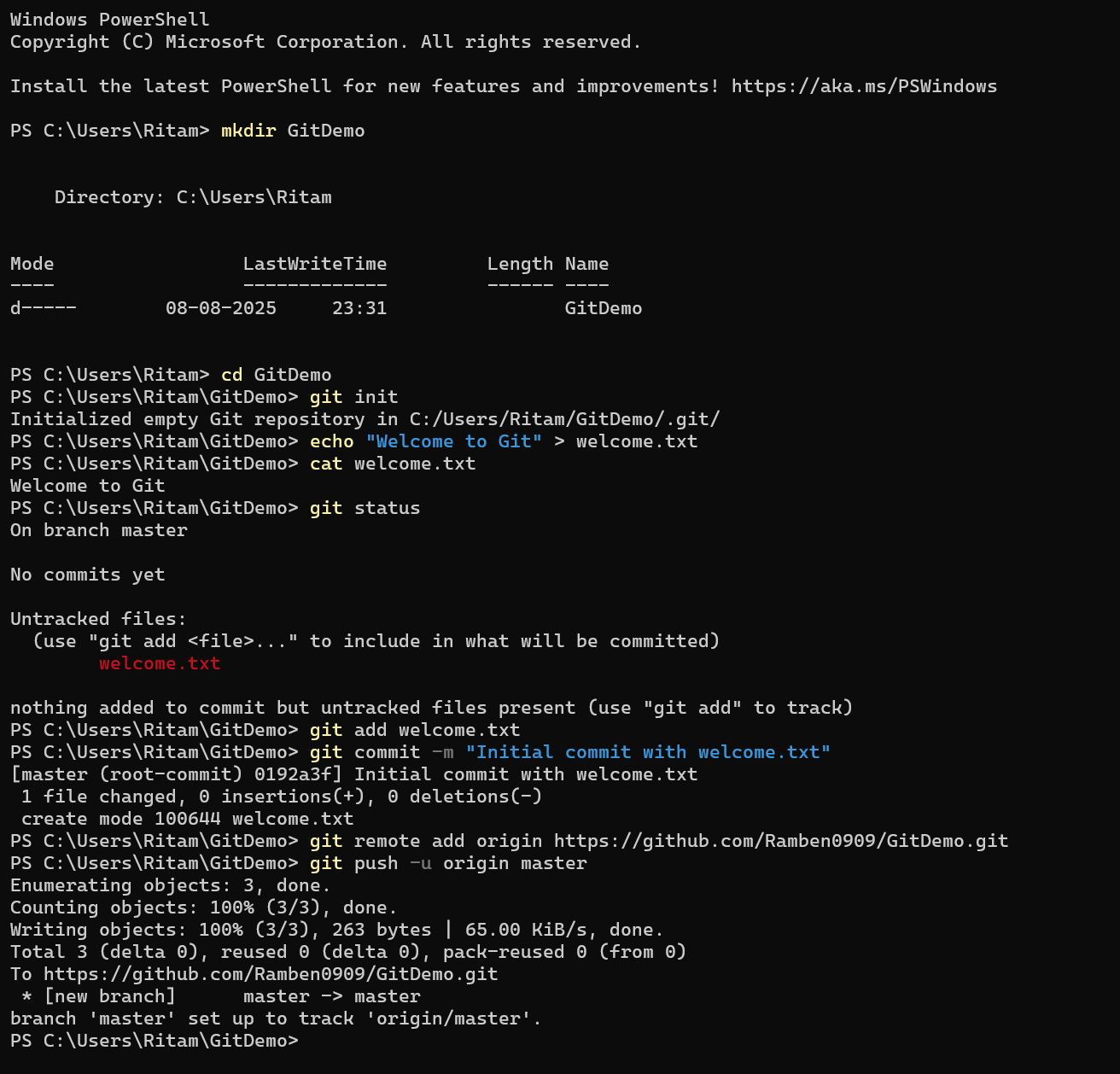
**Output**

## Part 2: Git Initialization, File Creation, Commit, and Push

Commands Executed:

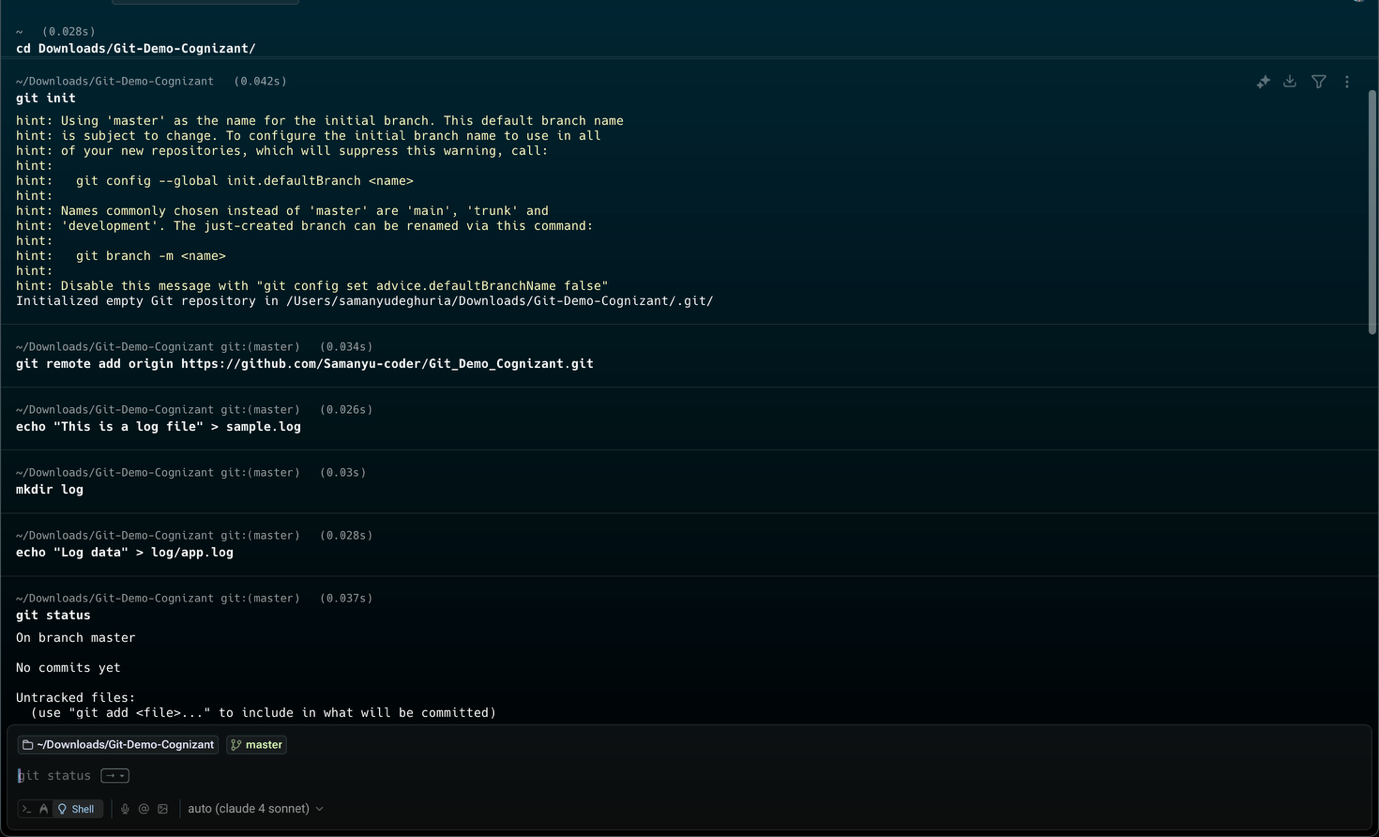
* mkdir GitDemo
* cd GitDemo
* git init
* echo "Welcome to Git" > welcome.txt
* cat welcome.txt
* git status
* git add welcome.txt
* git commit -m "Initial commit with welcome.txt"
* git remote add origin https://github.com/Ramben0909/GitDemo.git
* git push -u origin master

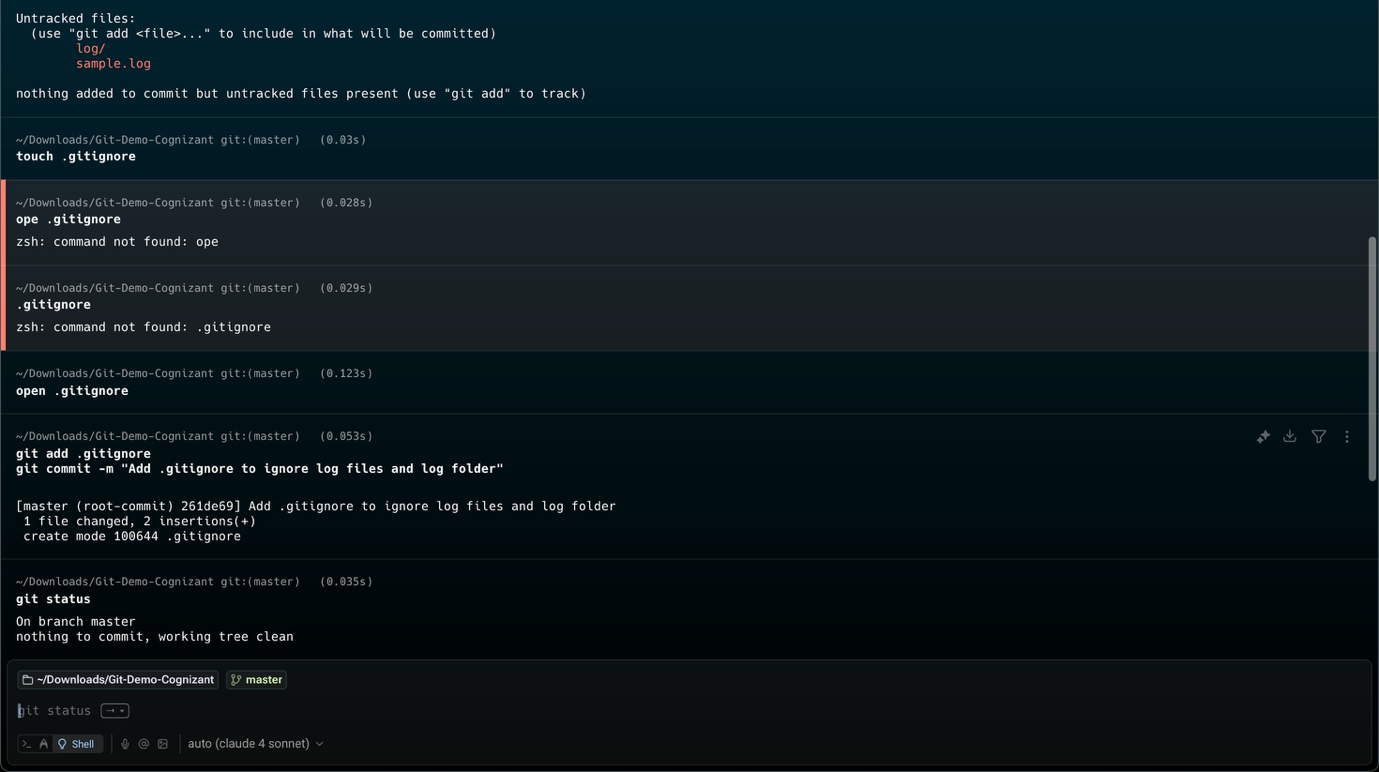
**Output:**



**Git HOL - 2**

**Output:**

****



A screenshot of a computer

AI-generated content may be incorrect.

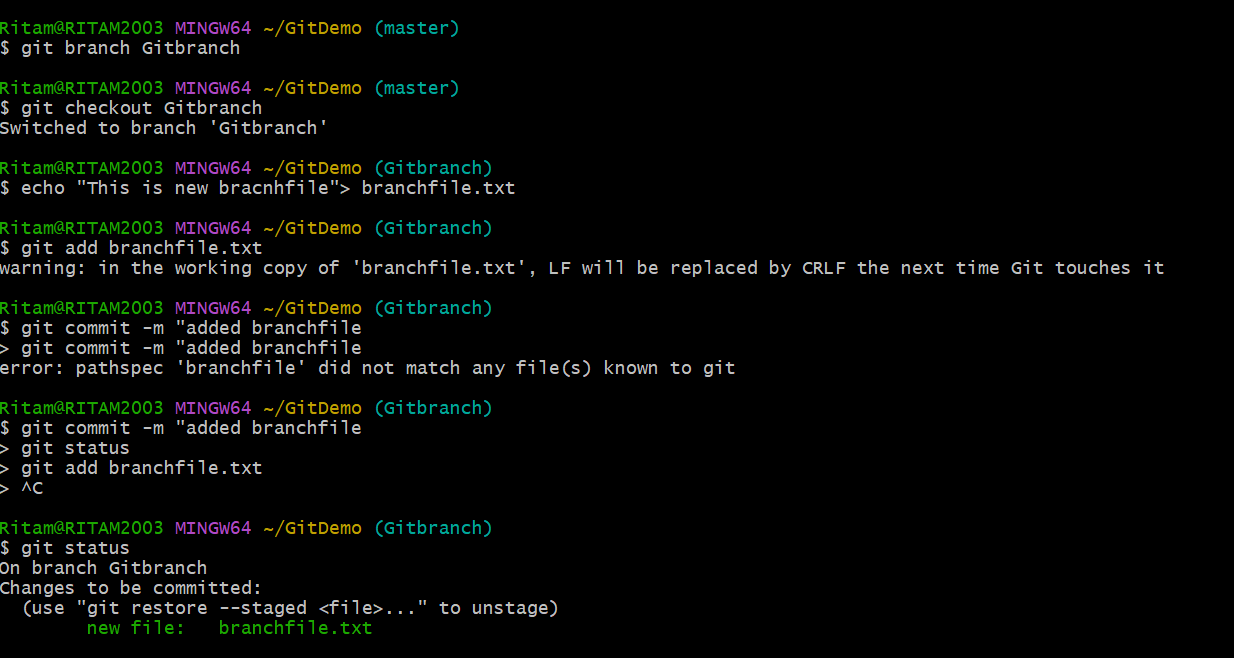
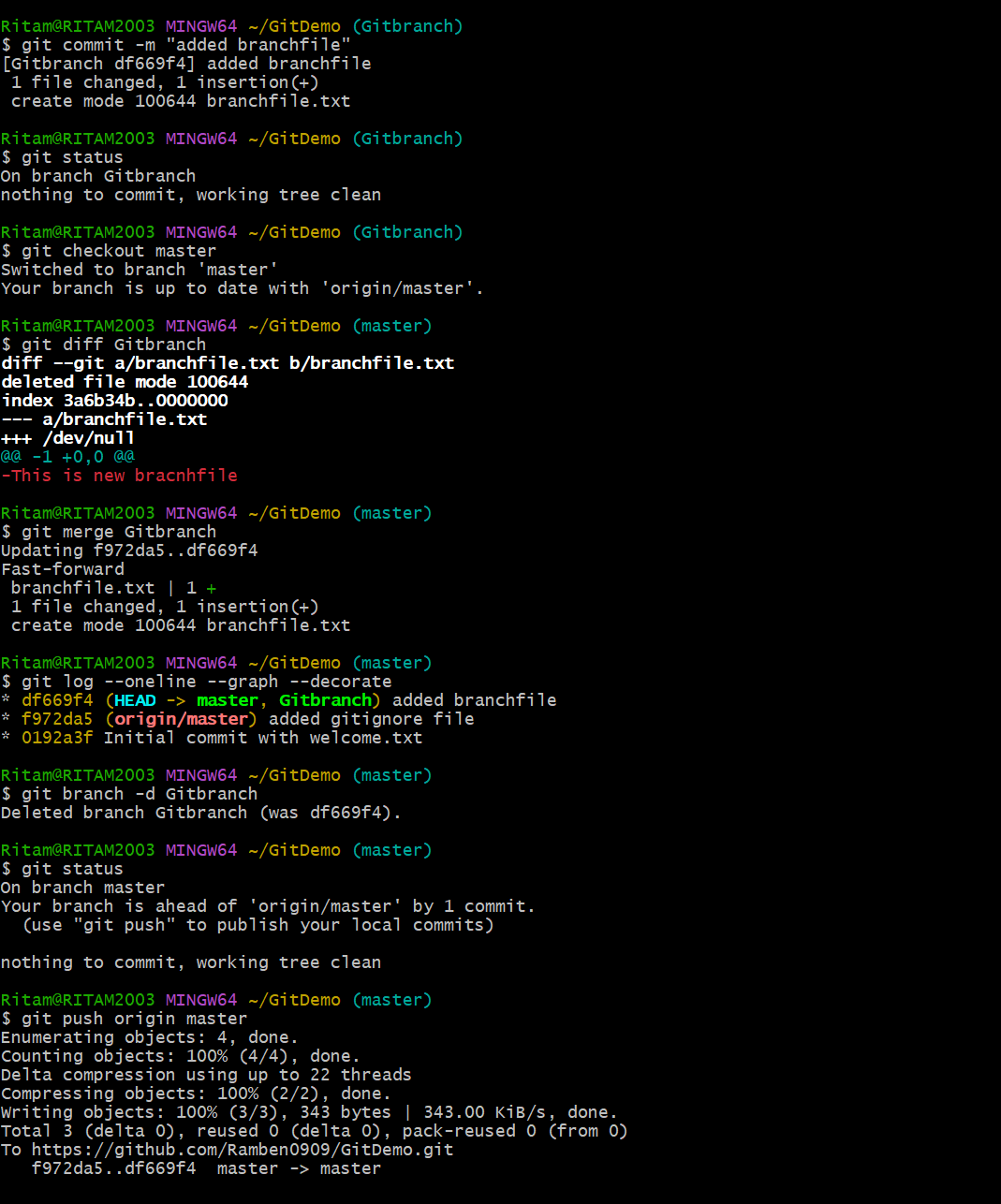
A screenshot of a computer

AI-generated content may be incorrect.

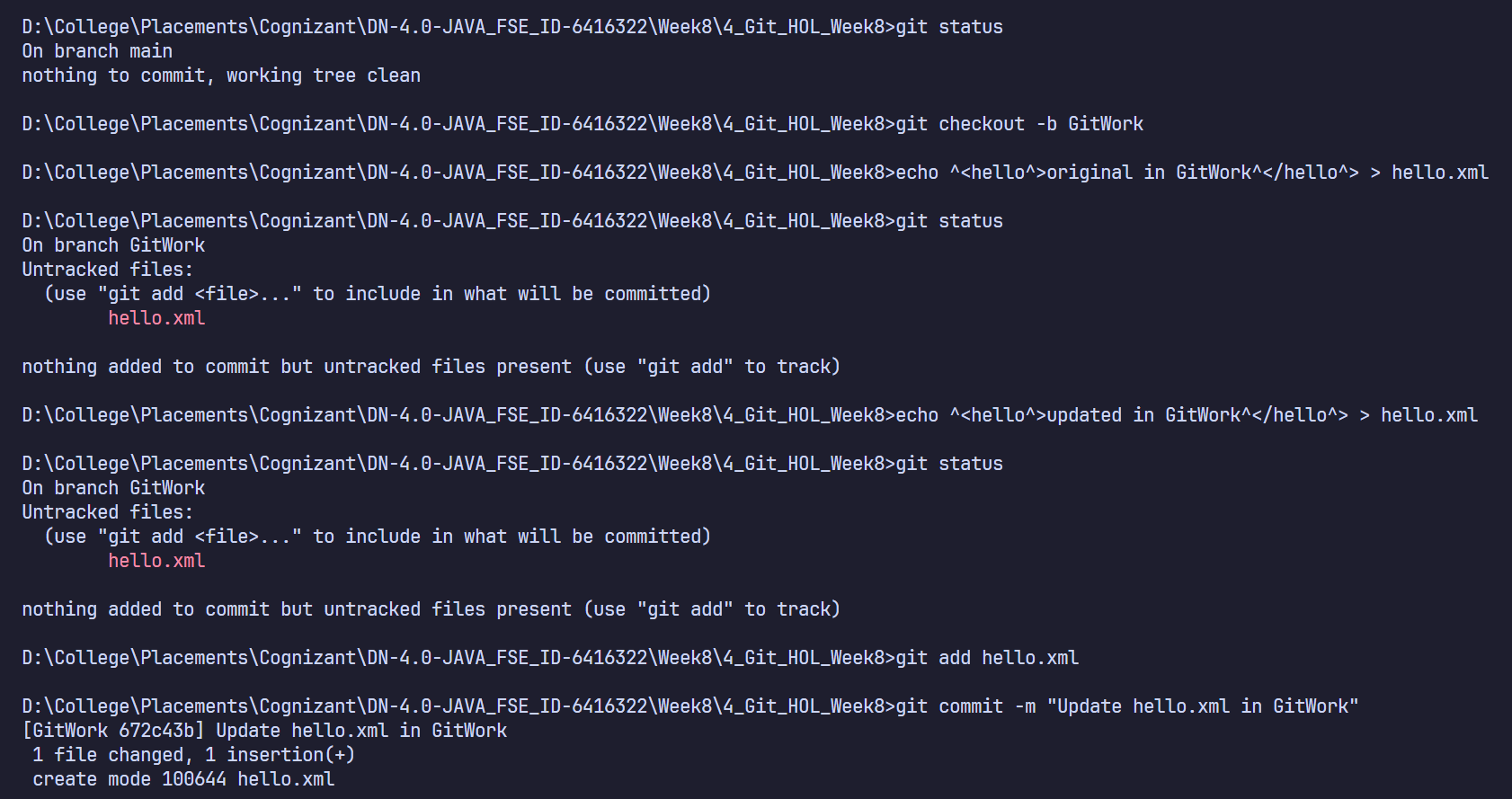
**Git HOL - 3**

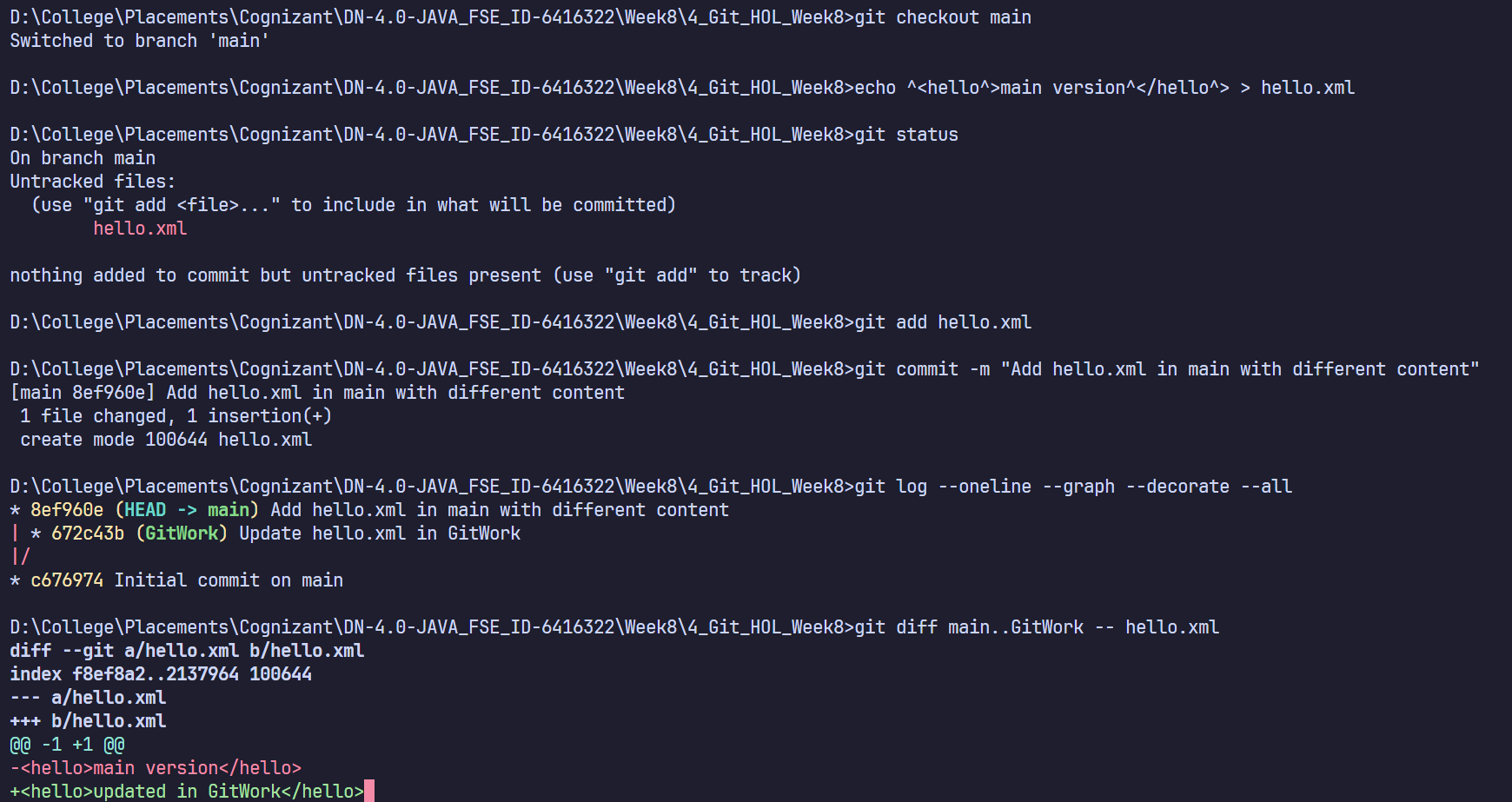
**Branch Creation, Commit, Merge, and Push**

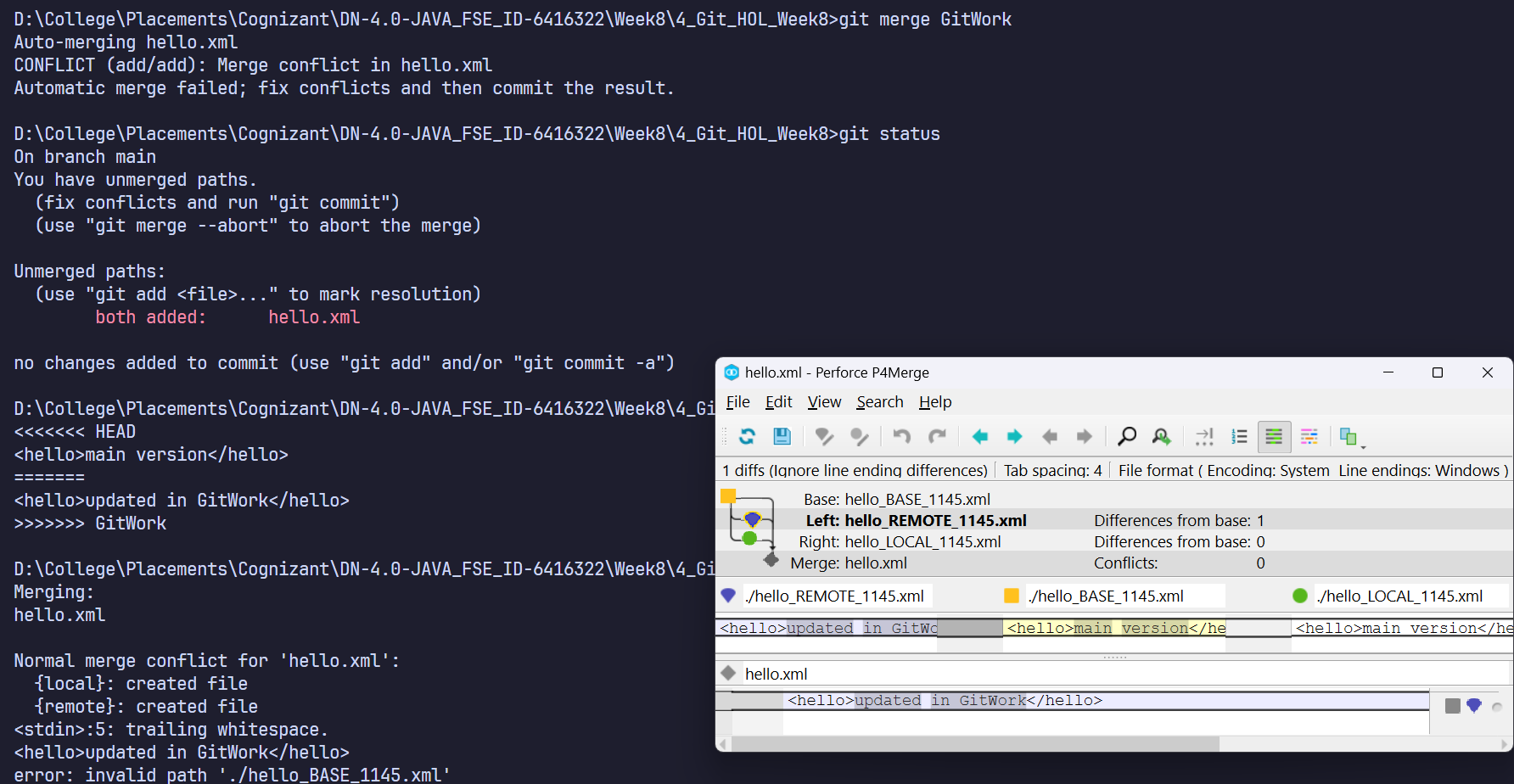
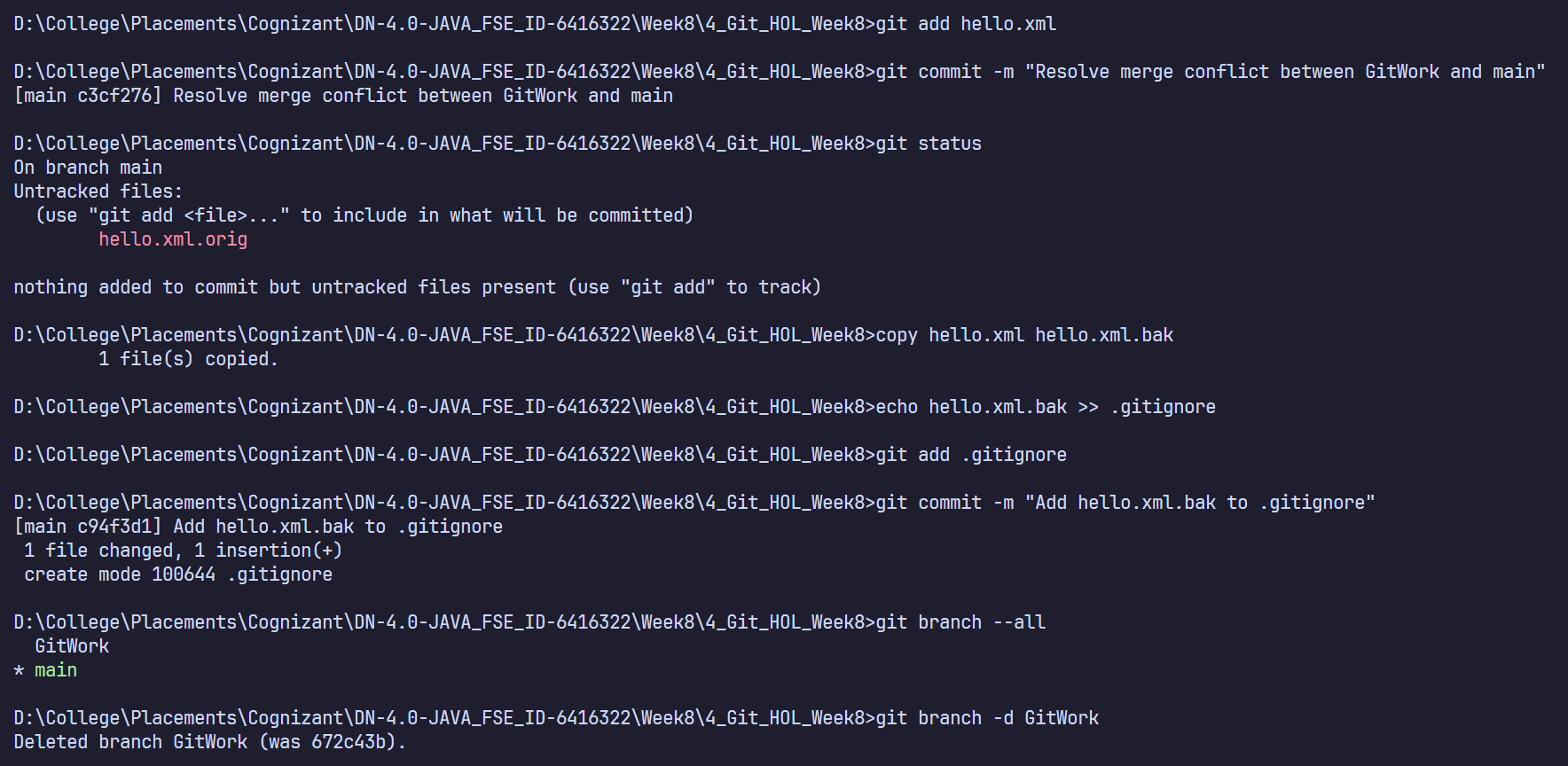
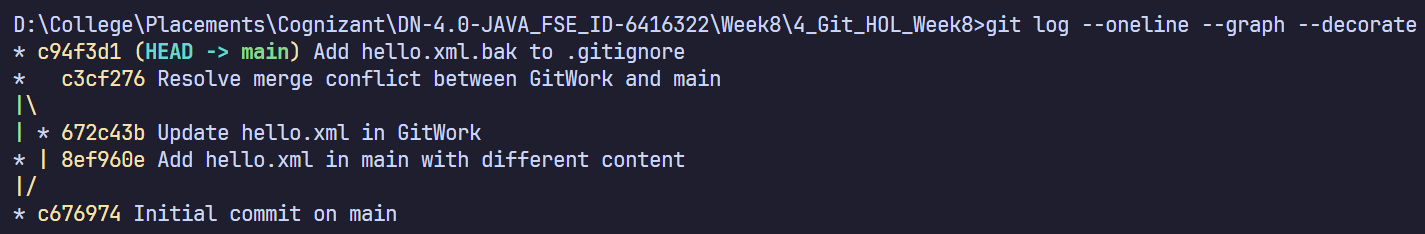
**Commands Executed**

* git branch Gitbranch
* git checkout Gitbranch
* echo "This is new branchfile" > branchfile.txt
* git add branchfile.txt
* git commit -m "added branchfile"
* git status
* git checkout master
* git diff Gitbranch
* git merge Gitbranch
* git log --oneline --graph --decorate
* git branch -d Gitbranch
* git status
* git push origin master

**Git HOL - 4**

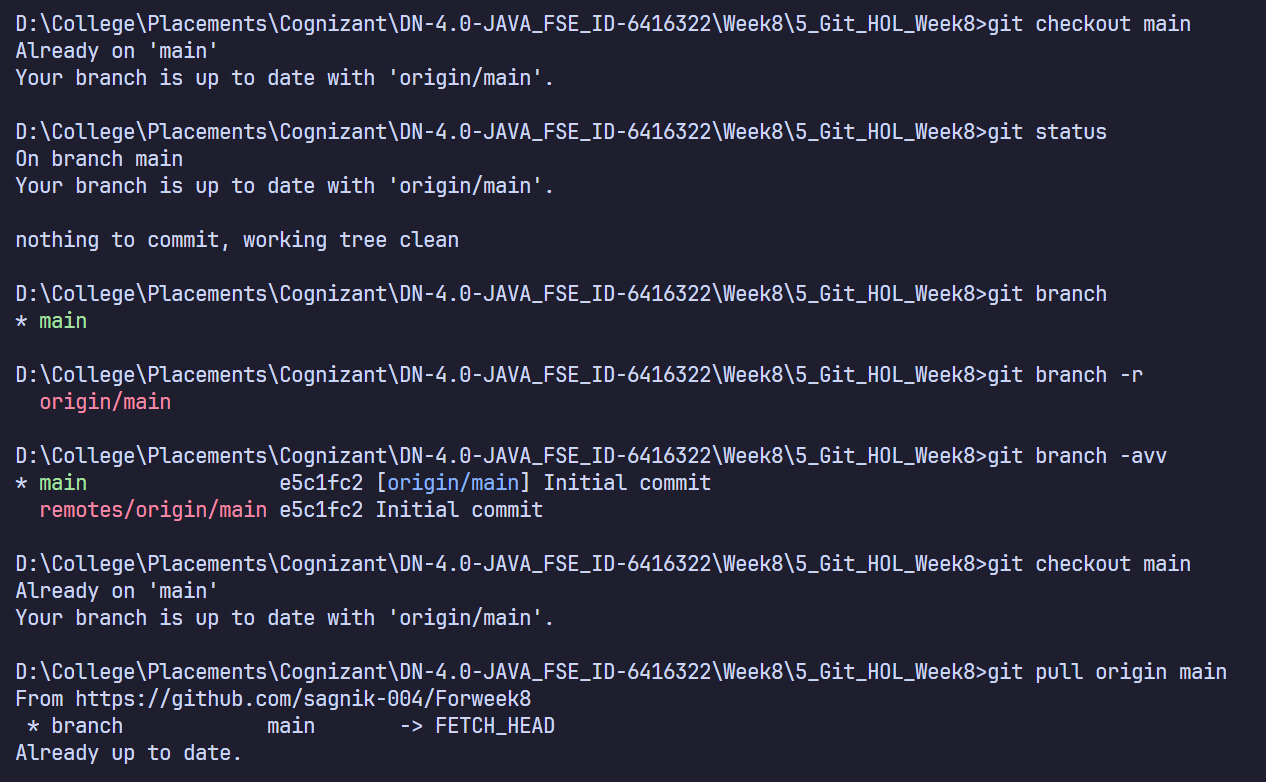
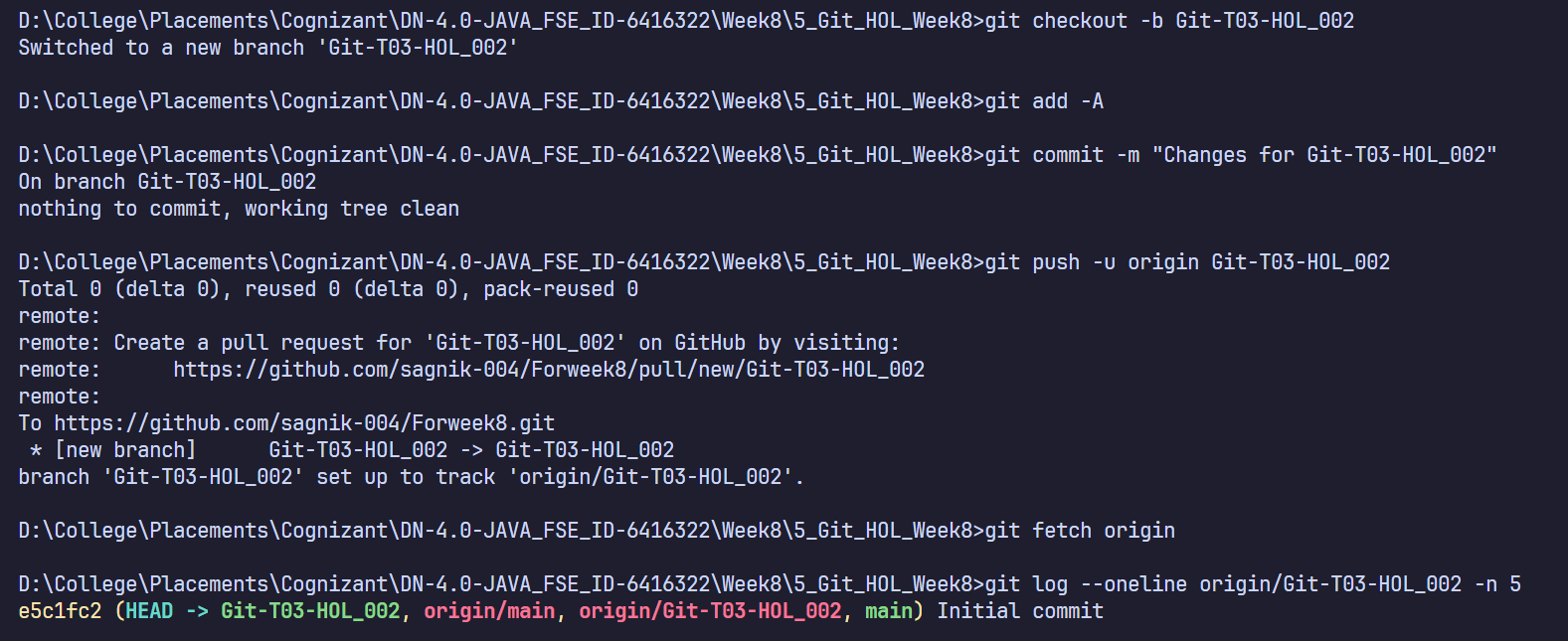
1. Verify if master is in clean state.
2. Create a branch **“GitWork”.** Add a file “hello.xml”.
3. Update the content of “hello.xml” and observe the status
4. Commit the changes to reflect in the branch
5. Switch to master.
6. Add a file **“hello.xml”** to the master and add some different content than previous.
7. Commit the changes to the master
8. Observe the log by executing **“git log –oneline –graph –decorate –all”**
9. Check the differences with Git diff tool
10. For better visualization, use P4Merge tool to list out all the differences between master and branch
11. Merge the branch to the master
12. Observe the git mark up.
13. Use 3-way merge tool to resolve the conflict
14. Commit the changes to the master, once done with conflict
15. Observe the git status and add backup file to the .gitignore file.
16. Commit the changes to the .gitignore
17. List out all the available branches
18. Delete the branch, which merge to master.
19. Observe the log by executing **“git log –oneline –graph –decorate”**





**Git HOL - 5**

1. Verify if master is in clean state.
2. List out all the available branches.
3. Pull the remote git repository to the master
4. Push the changes, which are pending from **“Git-T03-HOL\_002”** to the remote repository.

Observe if the changes are reflected in the remote repository.