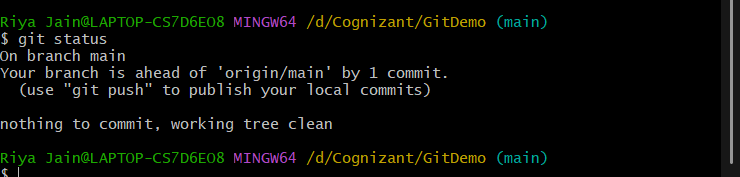
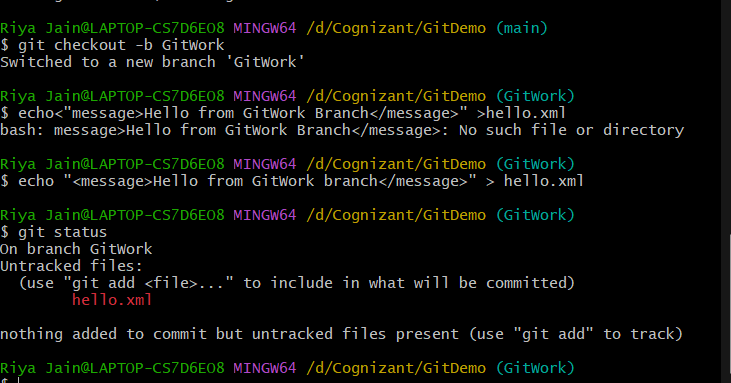
**Exercise – 4**

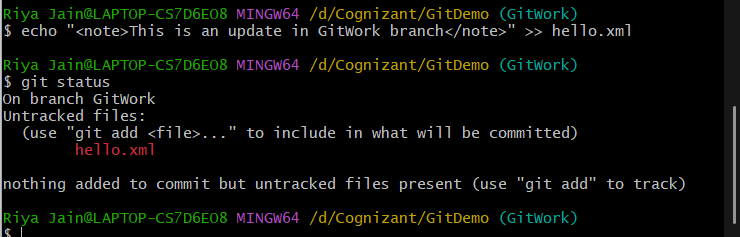
* **Step 1: Verify if master/main is in clean state.**

****

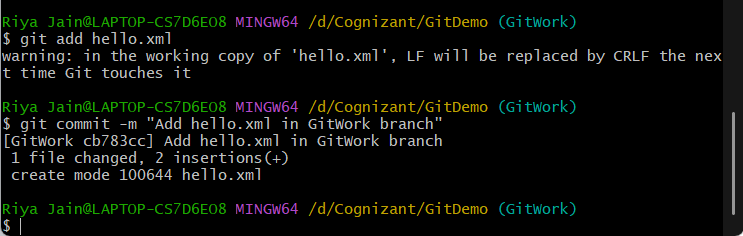
* **Step 2: Create a branch “GitWork”. Add a file “hello.xml”.**

****

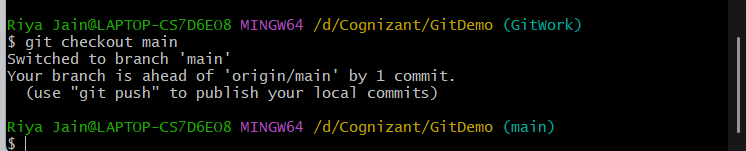
* **Step 3: Update the content of “hello.xml” and observe the status.**

****

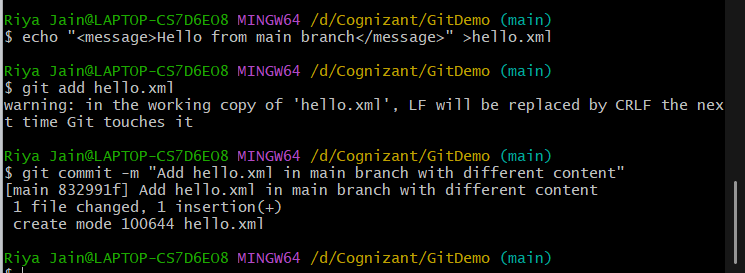
* **Step 4: Commit the changes to reflect in the branch.**

****

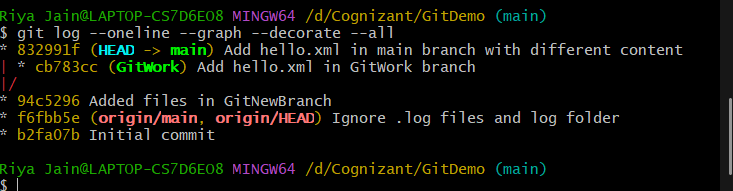
* **Step 5: Switch to master/main.**

****

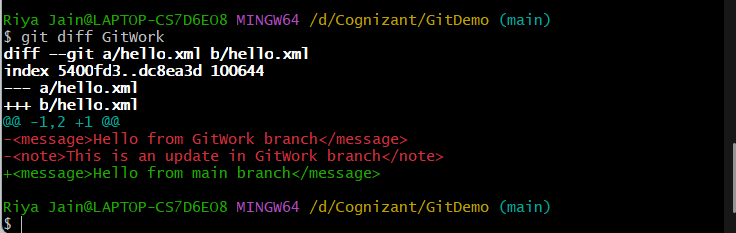
* **Step 6: Add a file “hello.xml” to the master/main and add some different content than previous.**
* **Step 7: Commit the changes to the master/main.**

****

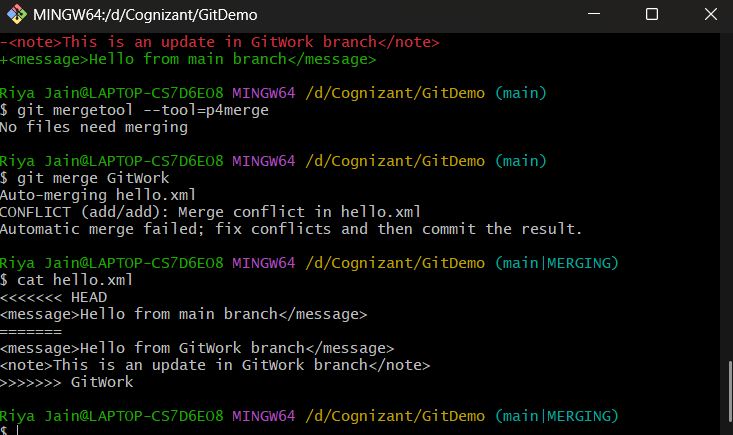
* **Step 8: Observe the log by executing “git log –oneline –graph –decorate –all”.**

****

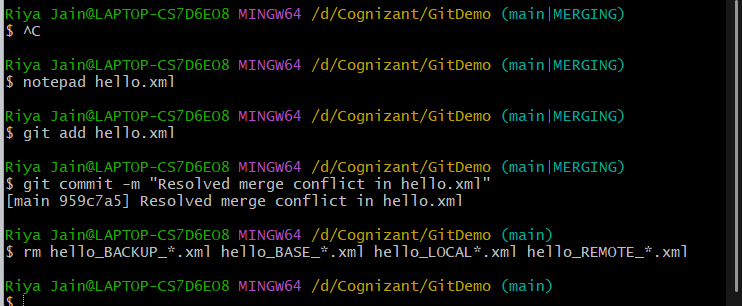
* **Step 9: Check the differences with Git diff tool.**

****

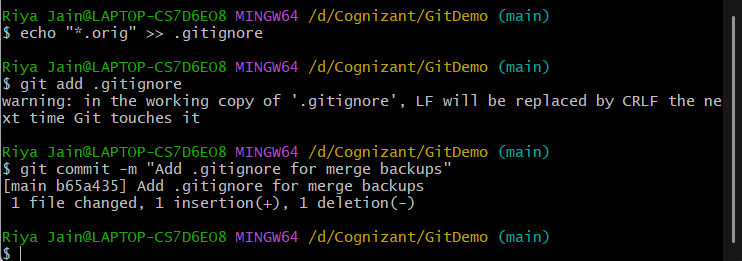
* **Step 10: For better visualization, use P4Merge tool to list out all the differences between master/main and branch.**
* **Step 11: Merge the branch to the master/main.**
* **Step 12: Observe the git mark up.**

****

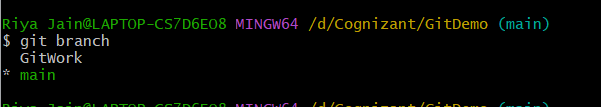
* **Step 13: Use 3-way merge tool to resolve the conflict.**
* **Step 14: Commit the changes to the master, once done with conflict .**

****

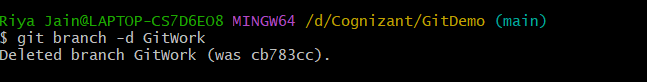
* **Step 15: Observe the git status and add backup file to the .gitignore file.**
* **Step 16: Commit the changes to the .gitignore.**

****

* **Step 17: List out all the available branches.**

****

* **Step 18: Delete the branch, which merge to master/main.**

****

* **Step 19: Observe the log by executing “git log –oneline –graph –decorate”.**

