**Lesson 02 Demo 07**

**Resolving Merge Conflicts in Git**

**Objective:** To demonstrate how to resolve merge conflicts in Git to maintain a clean codebase and project continuity when conflicting changes occur

**Tools required:** Git and GitHub

**Prerequisites:** Create a Git repository; refer Lesson 02 Demo 01

Steps to be followed:

1. Create a new branch
2. Make changes to the new branch
3. Switch to the main branch and make conflicting changes
4. Merge the new branch and resolve the merge conflicts

**Step 1: Create a new branch**

1. Execute the following commands to create and modify a new file in the main branch:

**touch example.txt**

**echo "Line 1: Hello, this is an example file. Here we can write text.”> example.txt**

****

1. Run the following commands to add and commit the changes:

**git add example.txt**

**git commit -m "Add example file"**

A close-up of a website

Description automatically generated

1. Run the following command to create and switch to a new branch:

**git checkout -b new-branch**

A close-up of a sign

Description automatically generated

**Step 2: Make changes to the new branch**

1. Execute the following commands to create a new file in the **new-branch:**

**echo "Line 2 from New-Branch : Hello, this is an example file." > example.txt**

**A black and white text with blue and green text

Description automatically generated**

1. Run the following commands to add and commit the changes:

**git add example.txt**

**git commit -m "Add example file"**

**A screenshot of a computer code

Description automatically generated**

**Step 3: Switch to the main branch and make conflicting changes**

1. Execute the following command to switch to the main branch:

**git checkout master**

A close-up of numbers

Description automatically generated

**echo " Line 2 from Master Branch: Hello, this is an example file." > example.txt**

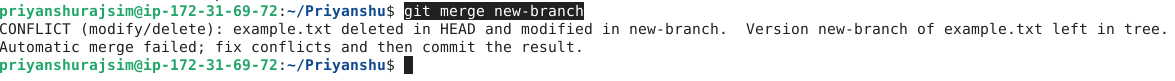
**git add example.txt**

**git commit -m "Add example file"**

**Step 4: Merge the new branch and resolve the merge conflicts**

1. Use the command below to merge the new branch into the main branch:

**git merge new-branch**

****

**Note:** Altering the same file differently on two branches will result in a merge conflict. To resolve it, open the conflicted file in a text editor and choose which changes to retain.

1. After editing, add the resolved file to the staging area using the following command:

**git add example.txt**

****

1. Commit the resolved changes:

**git commit -m "Resolved merge conflict"**

****

By following these steps, you have successfully resolved a merge conflict caused in Git.