**Assignments Swapnil Gaikwad**

1. What is a merge conflict in Git, and when does it typically occur?

Answer –

A merge conflict happens when Git cannot automatically merge changes because different changes were made to the same part of a file in two branches. It typically occurs when two developers edit the same line of code or make conflicting changes to the same file.

1. Why is it important to resolve merge conflicts before continuing with a merge?

Answer –

It ensures the code is consistent and functional. Git pauses the merge process until the conflict is resolved to prevent incomplete or broken code from being merged.

1. Describe the steps to resolve a merge conflict in Git.

Answer –

Identify the conflicting files (Git will highlight them after a merge attempt).

Open the files with conflicts and manually resolve them by choosing the correct changes.

Remove the conflict markers (<<<<<<<, =======, >>>>>>>).

Stage the resolved files using git add.

Complete the merge by running git commit.

1. What command do you use to check which files have conflicts in your repository?

Answer –

git status

1. If you encounter a merge conflict, how would you decide which changes to keep?

Answer –

Review both changes carefully to determine which version is correct or combine them if needed. This decision depends on the context of the code and project requirements.

1. Can you explain what the conflict markers (<<<<<<<, =======, >>>>>>>) represent in a conflicting file?

Answer –

<<<<<<<: Marks the beginning of your branch's changes.

=======: Separates your changes from the incoming changes.

>>>>>>>: Marks the end of the incoming branch's changes.

You need to edit the file to remove these markers and keep the appropriate changes.