COGNIZANT

HANDS ON

NAME: CIDHAM SK
EMP.ID: 906004
COHORT-CODE: INTADM21AJ009

\

1. **What is the difference between Git and SVN?**

Git	SVN
➢ Git is open source distributed vice control system developed by Linus Torvalds in 2005. It emphasis on speed and data integrity	➢ Apache Subversion is an open source software version and revision control system under Apache license.
➢ Git has a Distributed Model.	➢ SVN has a Centralized Model.
➢ In git every user has their own copy of code on their local like their own branch.	➢ In SVN there is central repository has working copy that also make changes and committed in central repository.
➢ In git we do not required any Network to perform git operation.	➢ In SVN we required Network for runs the SVN operation.

2. **Name a few Git commands and explain their usage?**

- **git init:**

This command turns a directory into an empty Git repository. This is the first step in creating a repository. After running git init, adding and committing files/directories is possible.

Usage:

```
# change directory to codebase
$ cd /file/path/to/code
# make directory a git repository
$ git init
```

- **git add:**

Adds files in the to the staging area for Git. Before a file is available to commit to a repository, the file needs to be added to the Git index (staging area). There are a few different ways to use git add, by adding entire directories, specific files, or all unstaged files.

Usage:

$ git add <file or directory name>

- **git commit:**

Record the changes made to the files to a local repository. For easy reference, each commit has a unique ID.

Usage:

Adding a commit with message

$ git commit -m "Commit message in quotes"

- **git status:**

This command returns the current state of the repository.

Usage:

$ git status

- **git config:**

With Git, there are many configurations and settings possible. git config is how to assign these settings. Two important settings are user user.name and user.email. These values set what email address and name commits will be from on a local computer. With git config, a --global flag is used to write the settings to all repositories on a computer. Without a --global flag settings will only apply to the current repository that you are currently in.

Usage:

$ git config <setting> <command>

3. **What is the function of ‘git config’?**

The git config command is a convenience function that is used to set Git configuration values on a global or local project level. These configuration levels correspond to. gitconfig text files. Executing git config will modify a configuration text file.

4. **Explain the different points when a merge can enter a conflicted stage. What is the difference between fork, branch, and clone?**

- Merging and conflicts are a common part of the Git experience. Conflicts in other version control tools like SVN can be costly and time-consuming. Git makes merging super easy. Most of the time, Git will figure out how to automatically integrate new changes. Conflicts generally arise when two people have changed the same lines in a file, or if one developer deleted a file while another developer was modifying it. In these cases, Git cannot automatically determine what is correct. Conflicts only affect the developer conducting the merge, the rest of the team is unaware of the conflict. Git will mark the file as being conflicted and halt the merging process. It is then the developers' responsibility to resolve the conflict.

Forking is a concept while cloning is a process. Forking is just containing a separate copy of the repository and there is no command involved. Cloning is done through the command ‘git clone ‘and it is a process of receiving all the code files to the local machine. Branching and forking provide two ways of diverging from the main code line. Both Mercurial and Git have the concept of branches at the local level. A repository code branch, like a branch of a tree, remains part of the original repository. The code that is branched (main trunk) and the branch know and

rely on each other. Like a tree trunk's branch, a code branch knows about the trunk (original code base) it originated from.

5. **What is the difference between rebasing and merge in Git?**

- git merge applies all unique commits from branch A into branch B in one commit with final result. It doesn’t rewrite commit history, just adds one new commit

- git rebase gets all unique commits from both branches and applies them one by one. It rewrites commit history but doesn’t create extra commit for mergin.