**Basic Git Questions**

* What is Git and why is it used?

**solution** :- Git is a distributed version control system (DVCS) used to track changes in source code during software development.

To manage and track changes in code over time.

To enable collaboration among multiple developers.

To create branches for new features or bug fixes.

To maintain a history of changes and revert to previous versions if needed.

* Explain the difference between Git and GitHub.

**solution** :-

Git:

A version control system used to track changes in files.

Works locally on your machine.

GitHub:

A cloud-based platform that hosts Git repositories.

Provides a graphical interface for managing Git repositories.

Enables collaboration through features like pull requests, issues, and code reviews.

* How do you install Git on your machine?

**solution** :-

Download the installer from git-scm.com.

Run the installer and follow the prompts.

* How do you configure your username and email in Git?

**solution** :-

git config --global user.name "Your Name"

git config --global user.email "your.email@puropale.com"

* What is a repository in Git?

**solution** :-

A repository (or repo) is a storage location for your project's files and their complete history of changes.

It contains all the commits, branches, and configuration data.

* How do you create a new Git repository?

**solution** :-

cd /path/to/your/project

git init

* How do you clone a repository from GitHub?

**solution** :-git clone <repository\_url>

* What is the purpose of the .gitignore file?

**solution** :-

The .gitignore file specifies files and directories that Git should ignore (e.g., temporary files, logs, or sensitive data).

* How do you check the status of your working directory in Git?

**solution** :-git status

* How do you add files to the staging area in Git?

**solution** :-git add <file\_name> or git add .

**Intermediate Git Questions**

* Explain the concept of commits in Git.

**solution** :-A commit is a snapshot of your project at a specific point in time.

It records changes to the repository and includes a commit message describing the changes.

* How do you create a new commit in Git?

**solution** :- git add .

git commit -m "Your commit message"

* What is the purpose of the git log command?

**solution** :-The git log command displays the commit history of the repository.

* How do you view the history of commits in a repository?

**solution** :-git log

* How do you view the changes made in a commit?

**solution** :-git show <commit\_hash>

* What is branching in Git and why is it useful?

**solution** :-Branching allows you to create separate lines of development within a repository.

Developing new features without affecting the main codebase.

Fixing bugs in isolation.

Experimenting with new ideas.

* How do you create a new branch in Git?

**solution** :- git branch <branch\_name>

* How do you switch between branches in Git?

**solution** :- git checkout <branch name>

* What is the difference between git merge and git rebase?

**solution** :-

git merge:

Combines changes from one branch into another.

Creates a new merge commit.

Preserves the history of both branches.

git rebase:

Moves or "replays" commits from one branch onto another.

Creates a linear history.

Avoids merge commits.

* How do you resolve merge conflicts in Git?

**solution** :-Open the conflicting file(s) and resolve the conflicts manually.

Mark the conflicts as resolved:

git add <file\_name>

Complete the merge:

git commit

**Git Exercises**

21.Create a new Git repository and configure your username and email.

**solution** :-mkdir my-git-project

cd my-git-project

git init

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

22.Create a file, add some content to it, and commit the changes.

**solution** :-Create a new file:

touch example.txt

Add content to the file:

echo "Hello, Git!" > example.txt

Add the file to the staging area:

git add example.txt

Commit the changes:

git commit -m "Added example.txt with initial content"

23.Create a .gitignore file and add rules to ignore specific files and directories.

**solution** :-

24.Clone an existing repository from GitHub and make some changes.

**solution** :-

Clone a repository from GitHub:

git clone <repo-url>

cd repository

Make some changes:

Edit a file or create a new file.

Example:

echo "This is a new change" > newfile.txt

Add and commit the changes:

git add newfile.txt

git commit -m "Added newfile.txt with some content"

25.Create a new branch, make some changes, and switch back to the main branch.

**solution** :-

Create a new branch:

git branch feature-branch

Switch to the new branch:

git checkout feature-branch

Make some changes:

Edit a file or create a new file.

Example:

echo "This is a change in the feature branch" > feature-file.txt

Add and commit the changes:

git add feature-file.txt

git commit -m "Added feature-file.txt in feature-branch"

Switch back to the main branch:

git checkout main