WORK-CASE №1

1.Git is a distributed version control system that allows you to track changes in program code and collaborate on projects. The main function of Git is to store the history of changes in your program code, allowing you to simultaneously track different versions of files and determine who made specific changes and when.

Git offers a wide range of commands and operations:

- git init: This command initializes a new Git repository in the current directory and starts tracking changes.

- git clone [URL]: Use this command to copy a remote Git repository to your computer for further work.

- git add [files]: This operation includes the changes you made or new files in the index, ready to be committed.

- git commit -m "[comment]": With this command, you commit the changes to the specified files and add a short comment explaining the changes.

- git status: Use this command to check the status of your working directory to see if there are any uncommented changes, files that have already been indexed, and changes that are ready to be committed.

- git branch: Lists the branches in the repository and shows the current branch.

- git checkout [branch]: Switch to another branch to continue working.

- git merge [branch]: Merge the current branch with another branch to merge changes.

- git pull: Update a working copy from a remote repository, synchronizing it with the latest changes.

- git push: Sends saved changes to a remote server for saving and distribution.

- git log: Displays the history of commits in the repository, allowing you to investigate changes.

- git diff [files]: Shows the difference between two commits or the current working copy and the index.

- git remote: View the available remote repositories you are working with.

- git stash: Used to temporarily store changes without saving the commit.

- git reset [commit]: Move the current branch to the specified commit, discarding previous changes.

2. A "commit" is a concept used in version control systems, especially in the context of file version control systems such as Git. A commit is a means of recording changes to the file system and the history of a project.

As you work on a project, you make various changes to files (additions, deletions, code changes, etc.). When you are ready to save these changes and add them to the project history, you create a commit.

The commit contains a description of the changes you made and is stored along with all the changes you made to the files at the time of the commit. Each commit has a unique identifier that allows you to track it within the project history.

This mechanism allows you to effectively track all changes in the project, restore previous project states, and collaborate on the project with the team, while maintaining the order and logic of the project.