LSS REMIDIAL

28/03/22
TUESDAY

Question: 01

Explain Git, how git works and how to configure Git and push the files to Git ~~to~~ Explain with commands.

Ans: Git is free and open source version control system designed to store files by making repository. for large or small projects. user can share his work with different users and rest of the members can modify the files and commit them whenever required.

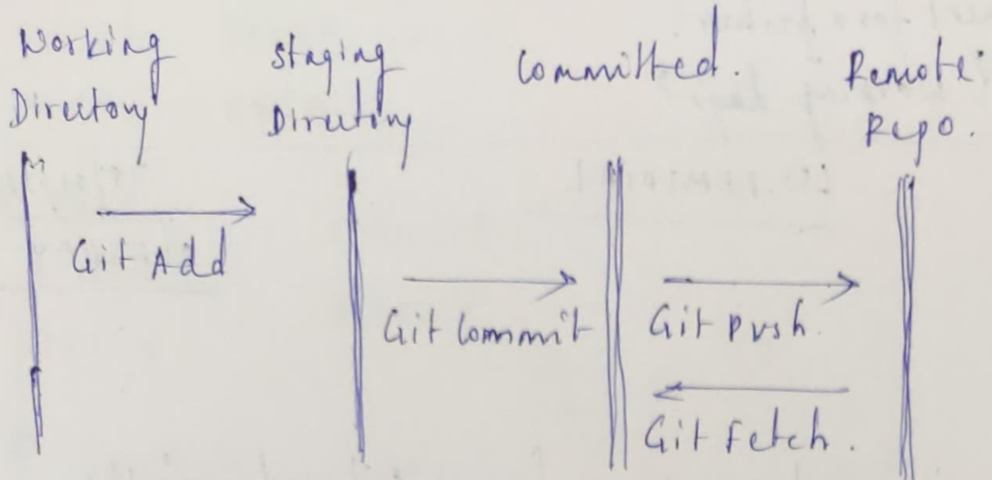
- Git allows user to store the files locally or store them on a website like Github. Users can commit changes to their repos or can instantly rollback to previous version of the file if the new file has some issues.

- Git projects go through several steps:

① Working directory: the user is working on his repo, commits are not yet tracked and not yet saved.

②. stagging directory: The files that are modified and now are ready for commit.

③. Committed: The snapshot of the files are stored as a commit history and users can ~~not~~ revert back to previous version of the file that he had saved.



⇒ Commands to ~~push~~ Configure and push files to Github:

```
$ git init.
```

Make a account of Github.

Intialize git

~~$~~ config your email and Github username.

```
$ git config --global user.name "____"
$ git config --global user.email "____"
```

→ make a directory with your username and prefix as git-

```
$ mkdir git-username.
```

→ Create some files or copy some files to created directory ~~to~~ so that we can push them to Github.

```
$ echo first > one.txt
$ echo second > two.txt.
```

→ You can check the status of the files, whether they are tracked by Github repo or not.

```
$ git status.
```

It will display the files and we havent pushed them to Github, so it will display it under untracked files.

→ Next we will add the created files to Github ~~staging~~ area, working directory

```
$ git add one.txt.
$ git add Two.txt.
```

Both files are added to staging area but are not yet committed.

→ Next we need to create a keygen using ed25519 encryption type associated with our Github email account.

```
$ ssh -keygen -t ed25519 -C "your-email"
```

→ Later we have to copy the created key-gen and add it to our Github account under settings > ssh keys.

→ To validate the generated key we will use ssh-agent.

```
$ eval 'ssh-agent -s'
~~$exec~~ $ exec ssh-agent bash.
$ sudo ssh-add vinay.
$ ssh-agent.
$ ssh -T @github.com
```

Now you have successfully authenticated with your github account.

→ Now re-initialize git.

```
$ git init.
$ git add readme.md.
$ git commit -m "first commit"
```

Now use your git remotely.

→ $ git remote add orgin "Github_repo_link"
$ git push -u origin main.

→ It will ask for username and password, enter them then your file will be pushed.

If it displays authentication failed then generate token in github and enter that token as password your files will be pushed to your Github.

$ Github > Developer settings > Generate personal token.
Generate it and copy it.

→ Execute the command $ git push -u orgin main again,
Enter your username,
now instead of your Github password enter the personal token as password.

∴ Your Files are been committed and pushed to Github.

Question: ② What are functions? Explain different types of functions with example.

Ans: Functions are used to perform set of instructions repeatedly that makes code reusability better.

Syntax for creating a function:

```
function_name () {
    list of commands.
  }.
```

Ex:

```
Hello () {        / Function declaration
  echo "Good Evening"
}
Hello   | calling the function.
```

2) Types of Function:

① Function with parameters:

Here we will pass two parameters with the function call, that parameters will be captured by the command and output will be generatted.

```
Ex: Hello () {
    ~~print~~ echo "Good Evening $1 $2"
    }
    Hello ABC XYZ.
```

② Recursive function:

Function that calls itself is known as recursive function. Other function can also call different function from the program.

```
Ex: Hello () {
    echo "Good Evening"
    Hello
    }.
```

Scope of variables:

You can use parameters like $1, $2, $@ to perform user defined parameters. In the echo statement mention them within " " and pass parameters with the function call.

```
Ex: Hello () {
      echo "Good morning $@"
    }
Hello Sir.
```