(Remedial) Assignment (Day-1)

⇒ git is used to ~~handle~~ tracking changes in the Source code Enabling multiple developers to ~~create~~ work together.

# Steps to add file on git through terminal:

i) ~~St~~ Install the git by using command "Sudo apt install git-all

ii) git clone address (where address is the URL given on the Software's git page.

iii) The Next Steup is to create a local repository on the System by using command "mkdir newproject"

iv) Now, we must initialize the repository in new project directory by issuing command git init.

v) Now, we have to add a file to the project by Using command "touch readme.txt".

vi) Check the Status by using command "git Status"

vii) and then add that file using git add readme.txt after that issue the "git Status" command again and can be Seen as readme.txt is now considered as a new file in the Project

viii) Now, we have to do our first commit (A commit is a record of files & that has beed Changed within the Project

ix) before commit we have to use the ~~com~~ following Commands to introduce

"git config -global user. email Piyush@gmail.com" "git config --global user.name "Piyush"

x) Now, we can create the commit by issuing the command:

"git commit -m "Descriptive message"

where descriptive message is the message about the changes within the commit for example, since this is the first commit for the readme.txt file the commit could be.

"git commit -m "first reading .txt file"

Creating a branch and pushing it to github.

i) git checkout -b BRANCH (where branch is the name of new branch).

ii) Next, we need to create a repository on GitHub. (on github account, click new Repo button from the account's main page).

iii) After creating the repository we have to use commands..

"git remote add origin URL"

"git Push -u origin master"

where URL is the url of our new github repository.

after using this command system will be prompted to authenticate username and password. once successfully authenticated the Project will be successfully pushed to our github repository.

Pulling the Project:

→ To pull the project files into local machine, issue the command.

"git pull origin master" (this command will pull down any new or changed files to local repository).

→ We can use another method also by generating the key using "ssh keygen" after commiting the file

→ ~~"ssh~~ "git log ssh- keygen -t ed25519 -C" Piyush@gmail"

→ a key will be generated and It will be saved as per the name given by the users.

→ gedit Piyush.pub (we have to use this command to copy the key

→ After that we have to use the command " eval 'ssh-agent -s'

→ Exec ssh-agent bash

→ ~~Exec ssh-agent ba~~ ssh-add Piyush

→ ssh -T git@github.com

→ echo "#mca" >> README.md,

→ git init

→ git add README.md

→ git commit -m "first commit"

→ git branch -M main

→ git remote add origin https: ________ .git

Now before pushing the file we have to generate tokens which will be done using following steps,

→ Click on setting in github page.

→ Click on Personal access tokens.

→ click on generate new tokens.

→ Copy the token to Enter it in place of Password

→ Use the command

"git push -u origin main

→ Username and password will be asked type username and In place of ~~token~~ Password Enter that token. your file will be Pushed to github.

Function

→ function Enable us to break down the overall functionality of a Script into Smaller, logical Sub Sections, Which can then be Called to perform their individual tasks When needed.

Creating a function :

To declare a function, we have to use the following Syntax-

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

Example:

```
Hello () {
    Echo " Hello. World"
}
Hello
```

Here, the name of the function is Hello.

O/p

```
Hello World.
```

Passing Parameters to a function:

We can define a function that will accept Parameters While calling it. These Paramenters would be represented $ Symbol.

Example:

```
Hii() {
    Echo " Hii, This $1 $2"
}
Hii This is ABC
```

OUTPUT: Hii This is ABC.

Returning values from functions:

If the user wants to terminate the Execution of function, then the return command uses there

the Syntax is: return ()

Example: To returns a value 10-

```
Hello () {
    Echo" Hello Hii $1 $2"
    return 10
}
Hello Hii ABC XYZ
Echo "Return value is $ret"
```

Here $ret is Captured Value returned by last command

ret= $.

Output:

```
Hello Hii ABC XYZ
Return value is 10
```

Nested function: A function that calls itself called nested function.

Example:

```
function1 (){
    Echo" This is first function"
    function2
}
```

```
function2 () {
        Echo" This is second function"
    }
# calling function1
      function1
```

Output:

```
This is first function
This is second function
```

function call from prompt:

In this we group the definitions in a file, say test.sh and then Execute the file in the current shell by typing

```
$. test.sh
```

This will lead to the function defined inside test.sh to be read and defined.