NAME : D . SAKETH

CLASS: CSE- A

ROLL NO: 21016

1. CREATING A REPOSITORY

2. CREATING FOLDER

```
veeru@SAKETH MINGW64 ~/cognizance (master)
$ git add task-1

veeru@SAKETH MINGW64 ~/cognizance (master)
$ git add task-1

veeru@SAKETH MINGW64 ~/cognizance (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   task-1/sample doc.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file_1.txt
        file_2.txt
        file_3.txt


veeru@SAKETH MINGW64 ~/cognizance (master)
$ git coomit -m "first commit"
git: 'coomit' is not a git command. See 'git --help'.

The most similar command is
        commit

veeru@SAKETH MINGW64 ~/cognizance (master)
$ git commit -m "first commit"
[master (root-commit) a9fb3b1] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 task-1/sample doc.txt

veeru@SAKETH MINGW64 ~/cognizance (master)
$ git branch -m main

veeru@SAKETH MINGW64 ~/cognizance (main)
$ git remote add origin https://github.com/Saketh0718/cognizance.git

veeru@SAKETH MINGW64 ~/cognizance (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Writing objects: 100% (4/4), 260 bytes | 130.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Saketh0718/cognizance.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

GIT ADD

```
veeru@SAKETH MINGW64 ~/cognizance (main)
$ git add file_1.txt  file_2.txt  file_3.txt
```

GIT CHECKOUT

```
veeru@SAKETH MINGW64 ~/cognizance (main)
$ git checkout main
Already on 'main'
A        file_1.txt
A        file_2.txt
A        file_3.txt
Your branch is up to date with 'origin/main'.
```

GIT CHECKOUT-B

```
" main

veeru@SAKETH MINGW64 ~/cognizance (main)
$ git checkout -b master
Switched to a new branch 'master'
```

GIT CLONE

```
veeru@SAKETH MINGW64 ~/cognizance (master)
$ git clone
fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

    -v, --verbose         be more verbose
    -q, --quiet           be more quiet
    --progress            force progress reporting
    --reject-shallow      don't clone shallow repository
    -n, --no-checkout     don't create a checkout
    --bare                create a bare repository
    --mirror              create a mirror repository (implies bare)
    -l, --local           to clone from a local repository
    --no-hardlinks        don't use local hardlinks, always copy
    -s, --shared          setup as shared repository
    --recurse-submodules[=<pathspec>]
                          initialize submodules in the clone
    --recursive ...       alias of --recurse-submodules
    -j, --jobs <n>        number of submodules cloned in parallel
    --template <template-directory>
                          directory from which templates will be used
    --reference <repo>    reference repository
    --reference-if-able <repo>
                          reference repository
    --dissociate          use --reference only while cloning
    -o, --origin <name>   use <name> instead of 'origin' to track upstream
    -b, --branch <branch>
                          checkout <branch> instead of the remote's HEAD
    -u, --upload-pack <path>
                          path to git-upload-pack on the remote
    --depth <depth>       create a shallow clone of that depth
    --shallow-since <time>
                          create a shallow clone since a specific time
    --shallow-exclude <revision>
                          deepen history of shallow clone, excluding rev
    --single-branch       clone only one branch, HEAD or --branch
    --no-tags             don't clone any tags, and make later fetches not to follow them
    --shallow-submodules  any cloned submodules will be shallow
    --separate-git-dir <gitdir>
                          separate git dir from working tree
    -c, --config <key=value>
                          set config inside the new repository
    --server-option <server-specific>
                          option to transmit
    -4, --ipv4            use IPv4 addresses only
    -6, --ipv6            use IPv6 addresses only
    --filter <args>       object filtering
    --remote-submodules   any cloned submodules will use their remote-tracking branch
    --sparse              initialize sparse-checkout file to include only files at root
```

GIT COMMIT

```
veeru@SAKETH MINGW64 ~/cognizance (master)
$ git commit -m "this is my new commit"
[master 0f3d347] this is my new commit
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file_1.txt
 create mode 100644 file_2.txt
 create mode 100644 file_3.txt
```

GIT PUSH

```
veeru@SAKETH MINGW64 ~/cognizance (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master
```

GIT RESET

```
veeru@SAKETH MINGW64 ~/cognizance (master)
$ git reset --hard
HEAD is now at 0f3d347 this is my new commit
```

GIT REBASE

```
veeru@SAKETH MINGW64 ~/cognizance (master)
$ git rebase
There is no tracking information for the current branch.
Please specify which branch you want to rebase against.
See git-rebase(1) for details.

    git rebase '<branch>'

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=<remote>/<branch> master
```

GIT PULL

```
veeru@SAKETH MINGW64 ~/cognizance (master)
$ git pull
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master
```

```
git config --list --show-origin
```

```
veeru@SAKETH MINGW64 ~/cognizance (main)
$ git config --list --show-origin
file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain
file:C:/Program Files/Git/etc/gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.process=git-lfs filter-process
file:C:/Program Files/Git/etc/gitconfig filter.lfs.required=true
file:C:/Program Files/Git/etc/gitconfig http.sslbackend=openssl
file:C:/Program Files/Git/etc/gitconfig http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true
file:C:/Program Files/Git/etc/gitconfig core.fscache=true
file:C:/Program Files/Git/etc/gitconfig core.symlinks=false
file:C:/Program Files/Git/etc/gitconfig core.usebuiltinfsmonitor=true
file:C:/Program Files/Git/etc/gitconfig pull.rebase=false
file:C:/Program Files/Git/etc/gitconfig credential.helper=manager-core
file:C:/Program Files/Git/etc/gitconfig credential.https://dev.azure.com.usehttppath=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/veeru/.gitconfig  user.email=dsaketh2004@gmail.com
file:C:/Users/veeru/.gitconfig  user.name=Saketh0718
file:.git/config       core.repositoryformatversion=0
file:.git/config       core.filemode=false
file:.git/config       core.bare=false
file:.git/config       core.logallrefupdates=true
file:.git/config       core.symlinks=false
file:.git/config       core.ignorecase=true
file:.git/config       remote.origin.url=https://github.com/Saketh0718/cognizance.git
file:.git/config       remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:.git/config       branch.main.remote=origin
file:.git/config       branch.main.merge=refs/heads/main
```

FORK