**Exercise 1 - Under The Hood of a Simple Commit**

C:\Users\HP\Desktop\Git Assignments>cd Advance\_Assignments

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git init

Initialized empty Git repository in C:/Users/HP/Desktop/Git Assignments/Advance\_Assignments/.git/

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

info.txt

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git add info.txt

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git commit -m"added new .txt file"

[master (root-commit) 3134aee] added new .txt file

1 file changed, 1 insertion(+)

create mode 100644 info.txt

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git log --graph --oneline --all

\* 3134aee (HEAD -> master) added new .txt file

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git cat-file

usage: git cat-file (-t [--allow-unknown-type] | -s [--allow-unknown-type] | -e | -p | <type> | --textconv | --filters) [--path=<path>] <object>

or: git cat-file (--batch | --batch-check) [--follow-symlinks] [--textconv | --filters]

<type> can be one of: blob, tree, commit, tag

-t show object type

-s show object size

-e exit with zero when there's no error

-p pretty-print object's content

--textconv for blob objects, run textconv on object's content

--filters for blob objects, run filters on object's content

--path <blob> use a specific path for --textconv/--filters

--allow-unknown-type allow -s and -t to work with broken/corrupt objects

--buffer buffer --batch output

--batch[=<format>] show info and content of objects fed from the standard input

--batch-check[=<format>]

show info about objects fed from the standard input

--follow-symlinks follow in-tree symlinks (used with --batch or --batch-check)

--batch-all-objects show all objects with --batch or --batch-check

--unordered do not order --batch-all-objects output

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git cat-file-t

git: 'cat-file-t' is not a git command. See 'git --help'.

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git cat-file -t

usage: git cat-file (-t [--allow-unknown-type] | -s [--allow-unknown-type] | -e | -p | <type> | --textconv | --filters) [--path=<path>] <object>

or: git cat-file (--batch | --batch-check) [--follow-symlinks] [--textconv | --filters]

<type> can be one of: blob, tree, commit, tag

-t show object type

-s show object size

-e exit with zero when there's no error

-p pretty-print object's content

--textconv for blob objects, run textconv on object's content

--filters for blob objects, run filters on object's content

--path <blob> use a specific path for --textconv/--filters

--allow-unknown-type allow -s and -t to work with broken/corrupt objects

--buffer buffer --batch output

--batch[=<format>] show info and content of objects fed from the standard input

--batch-check[=<format>]

show info about objects fed from the standard input

--follow-symlinks follow in-tree symlinks (used with --batch or --batch-check)

--batch-all-objects show all objects with --batch or --batch-check

--unordered do not order --batch-all-objects output

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git cat-file -t --allow-unknown-type

usage: git cat-file (-t [--allow-unknown-type] | -s [--allow-unknown-type] | -e | -p | <type> | --textconv | --filters) [--path=<path>] <object>

or: git cat-file (--batch | --batch-check) [--follow-symlinks] [--textconv | --filters]

<type> can be one of: blob, tree, commit, tag

-t show object type

-s show object size

-e exit with zero when there's no error

-p pretty-print object's content

--textconv for blob objects, run textconv on object's content

--filters for blob objects, run filters on object's content

--path <blob> use a specific path for --textconv/--filters

--allow-unknown-type allow -s and -t to work with broken/corrupt objects

--buffer buffer --batch output

--batch[=<format>] show info and content of objects fed from the standard input

--batch-check[=<format>]

show info about objects fed from the standard input

--follow-symlinks follow in-tree symlinks (used with --batch or --batch-check)

--batch-all-objects show all objects with --batch or --batch-check

--unordered do not order --batch-all-objects output

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git log

commit 3134aee96efea64927722d667daa4567b471f0a3 (HEAD -> master)

Author: fahu26 <fahimshaikh2606@gmail.com>

Date: Thu Aug 1 14:28:21 2019 +0530

added new .txt file

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>

**Exercise 2- References**

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>git clone https://github.com/nnja/advanced-git-exercises.git advanced-git-exercises

Cloning into 'advanced-git-exercises'...

remote: Enumerating objects: 28, done.

remote: Total 28 (delta 0), reused 0 (delta 0), pack-reused 28

Unpacking objects: 100% (28/28), done.

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments>cd advanced-git-exercises

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git branch

\* master

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git branch exercise3

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git branch

exercise3

\* master

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git checkout exercise3

Switched to branch 'exercise3'

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git log

commit 43388fee19744e8893467331d7853a6475a227b8 (HEAD -> exercise3, origin/master, origin/exercise7, origin/exercise4, origin/exercise2, origin/HEAD, master)

Author: Nina Zakharenko <nina@nnja.io>

Date: Wed Oct 4 18:51:49 2017 -0700

Initial commit

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git show-ref

43388fee19744e8893467331d7853a6475a227b8 refs/heads/exercise3

43388fee19744e8893467331d7853a6475a227b8 refs/heads/master

43388fee19744e8893467331d7853a6475a227b8 refs/remotes/origin/HEAD

a198ae0f1cd37126e49408a17112a81cb4c42776 refs/remotes/origin/exercise10

43388fee19744e8893467331d7853a6475a227b8 refs/remotes/origin/exercise2

e348ebc1187cb3b4066b1e9432a614b464bf9d07 refs/remotes/origin/exercise3

43388fee19744e8893467331d7853a6475a227b8 refs/remotes/origin/exercise4

ff91b70c7f23de6b168dc60478f270e8e3f992a8 refs/remotes/origin/exercise5

4b2b90ec4f526139ca9c81e22174ebf5b9c56b52 refs/remotes/origin/exercise6

43388fee19744e8893467331d7853a6475a227b8 refs/remotes/origin/exercise7

88f6e2864bd0829c71654f1d19096f436a66ce07 refs/remotes/origin/exercise9

43388fee19744e8893467331d7853a6475a227b8 refs/remotes/origin/master

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git tag

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git show

commit 43388fee19744e8893467331d7853a6475a227b8 (HEAD -> exercise3, origin/master, origin/exercise7, origin/exercise4, origin/exercise2, origin/HEAD, master)

Author: Nina Zakharenko <nina@nnja.io>

Date: Wed Oct 4 18:51:49 2017 -0700

Initial commit

diff --git a/hello.txt b/hello.txt

new file mode 100644

index 0000000..980a0d5

--- /dev/null

+++ b/hello.txt

@@ -0,0 +1 @@

+Hello World!

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git tag -a

usage: git tag [-a | -s | -u <key-id>] [-f] [-m <msg> | -F <file>]

<tagname> [<head>]

or: git tag -d <tagname>...

or: git tag -l [-n[<num>]] [--contains <commit>] [--no-contains <commit>] [--points-at <object>]

[--format=<format>] [--[no-]merged [<commit>]] [<pattern>...]

or: git tag -v [--format=<format>] <tagname>...

-l, --list list tag names

-n[<n>] print <n> lines of each tag message

-d, --delete delete tags

-v, --verify verify tags

Tag creation options

-a, --annotate annotated tag, needs a message

-m, --message <message>

tag message

-F, --file <file> read message from file

-e, --edit force edit of tag message

-s, --sign annotated and GPG-signed tag

--cleanup <mode> how to strip spaces and #comments from message

-u, --local-user <key-id>

use another key to sign the tag

-f, --force replace the tag if exists

--create-reflog create a reflog

Tag listing options

--column[=<style>] show tag list in columns

--contains <commit> print only tags that contain the commit

Search skips displayed screen (press RETURN)

There is no --column option (press RETURN)

usage: git tag [-a | -s | -u <key-id>] [-f] [-m <msg> | -F <file>]

<tagname> [<head>]

or: git tag -d <tagname>...

or: git tag -l [-n[<num>]] [--contains <commit>] [--no-contains <commit>] [--points-at <object>]

[--format=<format>] [--[no-]merged [<commit>]] [<pattern>...]

or: git tag -v [--format=<format>] <tagname>...

-l, --list list tag names

-n[<n>] print <n> lines of each tag message

-d, --delete delete tags

-v, --verify verify tags

Tag creation options

-a, --annotate annotated tag, needs a message

-m, --message <message>

tag message

-F, --file <file> read message from file

-e, --edit force edit of tag message

-s, --sign annotated and GPG-signed tag

--cleanup <mode> how to strip spaces and #comments from message

-u, --local-user <key-id>

use another key to sign the tag

-f, --force replace the tag if exists

--create-reflog create a reflog

Tag listing options

--column[=<style>] show tag list in columns

--contains <commit> print only tags that contain the commit

--no-contains <commit>

print only tags that don't contain the commit

--merged <commit> print only tags that are merged

--no-merged <commit> print only tags that are not merged

--sort <key> field name to sort on

--points-at <object> print only tags of the object

--format <format> format to use for the output

--color[=<when>] respect format colors

-i, --ignore-case sorting and filtering are case insensitive

set mark: ...skipping...

-v, --verify verify tags

Tag creation options

-a, --annotate annotated tag, needs a message

-m, --message <message>

tag message

-F, --file <file> read message from file

-e, --edit force edit of tag message

-s, --sign annotated and GPG-signed tag

--cleanup <mode> how to strip spaces and #comments from message

-u, --local-user <key-id>

use another key to sign the tag

-f, --force replace the tag if exists

--create-reflog create a reflog

Tag listing options

--column[=<style>] show tag list in columns

--contains <commit> print only tags that contain the commit

--no-contains <commit>

print only tags that don't contain the commit

--merged <commit> print only tags that are merged

--no-merged <commit> print only tags that are not merged

--sort <key> field name to sort on

--points-at <object> print only tags of the object

--format <format> format to use for the output

--color[=<when>] respect format colors

-i, --ignore-case sorting and filtering are case insensitive

...skipping...

-v, --verify verify tags

Tag creation options

-a, --annotate annotated tag, needs a message

-m, --message <message>

tag message

-F, --file <file> read message from file

-e, --edit force edit of tag message

-s, --sign annotated and GPG-signed tag

--cleanup <mode> how to strip spaces and #comments from message

-u, --local-user <key-id>

use another key to sign the tag

-f, --force replace the tag if exists

--create-reflog create a reflog

Tag listing options

--column[=<style>] show tag list in columns

--contains <commit> print only tags that contain the commit

--no-contains <commit>

print only tags that don't contain the commit

--merged <commit> print only tags that are merged

--no-merged <commit> print only tags that are not merged

--sort <key> field name to sort on

--points-at <object> print only tags of the object

--format <format> format to use for the output

--color[=<when>] respect format colors

-i, --ignore-case sorting and filtering are case insensitive

/bin/sh: -c: line 0: unexpected EOF while looking for matching `"'

/bin/sh: -c: line 1: syntax error: unexpected end of file

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git tag -a v1.4 -m "my version 1.4"

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>git tag

v1.4

C:\Users\HP\Desktop\Git Assignments\Advance\_Assignments\advanced-git-exercises>

**Exercises 3: Forks And Remote Repos**