**Week 2- Day 5: Coding Challenge**

(Maximum marks -15)

AttainuID: #6

**Q-1) Explain the following commands in your own words:**

**1. git init**

**2. git status**

**3. git add**

**4. git commit**

**5. git push**

**6. git pull**

**7. git clone**

**8. git log**

**9. git diff**

**10. git checkout**

**Marks distribution:**

Question 1 carries 15 marks

1. **git init:**

It creates a empty git repository in the folder to track the files . After this

command .git folder can be seen in the directory. If it is already initialized

then again adding the command will reinitialize the git repository.

1. **git status:**

It shows all the tracked and untracked files in the directory. Tracked files will be in

green color and untracked will be indicated in red color.

1. **git add:**

Git add . - It will add all files in the directory to the staging area.

Git add filename - will add only the specified file to the staging area.

**4. git commit:**

**Git commit - m “message”** - It is used to created checkpoints, If any issue occurs using the commit id we can reverse back to that point, message followed by commit is used to see what changes has been made.

**5.git push:**

It is used to transfer or push the files or directories from local to the github repository. Pushing into master branch is not possible.

**Git push origin branch name -** destination branch needed to push into a specify branch.

1. **git pull :**

This command is used to fetch or get all the files from the github repository to the local directory If the repository cloned already.

If two or more developers are working in the same file changes done will be

added in the same directory so it will get the latest version of the file or directory.

1. **git clone :**

**Git clone url**

This command is used to get or clone the specified github repository to the local directory, all the files and folders from the repo will be added in the local.

1. **git log :**

This command maintains the history of all the commits that have made in the current branch, It is used to see all the previous commits.

1. **git diff:**

This command is used to many purposes. It will show or lists the difference between two files or two branches or two commits

**Git diff commitID1 commitID2**

**Git diff branch1 branch2**

**Git diff test.txt test.txt - if we edit a file after commit wanted to see the changes this command will show the difference.**

1. **git checkout:**

**Git checkout commit Id**

**Git checkout branchname**

It is used to jump from from commit to another or even from one branch to another branch. From this it will move to that particular checkpoint where adding or removing or any changes can be done.