**SWE 525 GIT Version Control Mid Term Exam 08/21/2016**

**ITU Student Name: Supriya Nittur Kumar**

**ITU Student ID: 89706**

1. Perform the following steps on any repository of your choice and push it to remote. Explain your work and paste your github repo address. Make me collaborator. Draw flow chart if applicable.
2. Create a repository

Answer: created a repository “midtermgit” on github and used git clone <https://github.com/Supriyankumar/midtermgit.git> which now has a local copy on my computer as midtermgit folder.

Last login: Sun Aug 21 09:27:01 on console

SUPRIYAs-MacBook-Pro:~ suppi$ cd desktop

SUPRIYAs-MacBook-Pro:desktop suppi$ cd SWE 525

-bash: cd: SWE: No such file or directory

SUPRIYAs-MacBook-Pro:desktop suppi$ SWE525

-bash: SWE525: command not found

SUPRIYAs-MacBook-Pro:desktop suppi$ cd SWE525

SUPRIYAs-MacBook-Pro:SWE525 suppi$ git clone https://github.com/Supriyankumar/midtermgit.git

Cloning into 'midtermgit'...

remote: Counting objects: 3, done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

SUPRIYAs-MacBook-Pro:SWE525 suppi$ cd midtermgit

SUPRIYAs-MacBook-Pro:midtermgit suppi$ ls

README.md

1. Make some changes -- add some files, change content of some files

Answer:

SUPRIYAs-MacBook-Pro:midtermgit suppi$ ls

README.md

SWE 525 GIT Version Control Mid Term Exam 08.docx

SUPRIYAs-MacBook-Pro:midtermgit suppi$ echo "#This is a new test file" > file1

SUPRIYAs-MacBook-Pro:midtermgit suppi$ ls

README.md

SWE 525 GIT Version Control Mid Term Exam 08.docx

file1

SWE 525 GIT Version Control Mid Term Exam 08.docx

file1

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git add file1

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git commit -m 'add new file1'

[master 84c148b] add new file1

1 file changed, 1 insertion(+)

create mode 100644 file1

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git status

# On branch master

# Your branch is ahead of 'origin/master' by 1 commit.

#

# Untracked files:

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

#

# SWE 525 GIT Version Control Mid Term Exam 08.docx

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

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git add file1

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git commit -m 'new line added'

[master cffb938] new line added

1 file changed, 1 insertion(+)

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git status

# On branch master

# Your branch is ahead of 'origin/master' by 2 commits.

#

# Untracked files:

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

#

# SWE 525 GIT Version Control Mid Term Exam 08.docx

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

1. Commit changes

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git commit -m 'add new file1'

[master 84c148b] add new file1

1 file changed, 1 insertion(+)

create mode 100644 file1

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git status

# On branch master

# Your branch is ahead of 'origin/master' by 1 commit.

#

# Untracked files:

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

#

# SWE 525 GIT Version Control Mid Term Exam 08.docx

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

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git add file1

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git commit -m 'new line added'

[master cffb938] new line added

1 file changed, 1 insertion(+)

SUPRIYAs-MacBook-Pro:midtermgit suppi$ git status

# On branch master

# Your branch is ahead of 'origin/master' by 2 commits.

#

# Untracked files:

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

#

# SWE 525 GIT Version Control Mid Term Exam 08.docx

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

4. repeat steps 1-3 for 2 more times. try to remove and ignore some files . Use git diff before each commit.

5. create changes and revert those changes using git when those changes are not added to staging area?

6. create changes and revert those changes after adding changes to staging area?

7. create change and revert those changes after committing those changes .

8. Check repository status before adding changes to staging area, after adding changes to staging area and after committing changes .

9. Display all the commits .

10. Display a specific commit .

11. create branches called "feature1" and "feature2" from master.

12. Delete branch "feature2".

13. Make some commits on "feature1".

14. Make some commits on branch "master".

15. merge branch "feature1" in "master" (make some conflicts by changing same file on same line in both the branches)

16. create branch "feature2" from master .

17. Make some commits on "feature2".

18. Make some commits on branch "master".

19. rebase branch "master" in "feature2" (make some conflicts by changing same file on same line in both the branches) .

20. find the differences between "merge" and "rebase" in above cases.

21. Use git stash

22. create a repository on github for your working repository and push changes on remote repository.