Software Project Management Lab 1

Using Git & GitHub

Introduction:

In this lab, we will be covering all basics regarding Git & GitHub. You will become familiar and comfortable with all essential Git commands and be capable of dealing with local and remote version control effectively.

Dealing with Git is not just essential for this course, as it will be used to submit your lab assignments, but is a crucial skill to have as a software engineer.

Lab Preparation:

1. In order to follow the steps of the lab, you will first need to install Git on your PC. Download Git from: https://git-scm.com/downloads
   Install using default settings

2. You will also need to create a GitHub account if you haven't already.
   Head over to: https://github.com/
   Sign up for GitHub

What is version control?

Version control is a system that basically manages code. In a typical project, multiple developers tend to work on their respective modules. Using version control, they can all "clone" the project "repository", work on their files, "commit" when wanted, "revert" if changes are unwanted and finally "push" their changes to the version control software that keeps track of all changes made to all files in the project.

There are two types of version control: local & remote. The point is to make a local version of the project where you will work on your own modules, then push your changes to the remote version of the project which contains different modules of other developers.

The system we'll be using is Git, which is a local version control software that allows you to create repositories and use different Git commands on

your local machine. To push our work online, we will be using GitHub, a remote version control software.

Git Command Cheat-Sheet:

git init #Initialize your git repository in the current directory

git config --global.name “Name” #Set your git name

git config --global.email “email@email.com” #Set your git email

git add file #Add file to staging area to be committed to repository, must be used whenever file is edited

git add *.txt #Adds all files with the .txt file extension (valid for any extension) git add . #Adds all files in current directory

git status #Check on the status of staging area

git rm --cached “file” #Remove file from repository, without removing from current directory

git clean -n #Check which untracked files are going to be removed entirely

git clean -f #Remove the untracked files

git commit -m “Some random comment” #Commit changes, adding a comment in the same command line

git add .getignore #add .getignore file which contains files to be ignored when commiting

git branch branchName #create a new branch seperate from master branch

git checkout branchName #work on the mentioned branch

git merge branchName #merge the changes made on the branch with master branch

git remote add origin www.(remote repository link).com #add remote remote repository defined as “origin”

git remote #check added remote repositories

git push -u origin master #push all local changes to remote repository defined as “origin” git clone www.(remote repository link).com #Download and make a copy of remote repository on local machine

git pull #Download any changes made to the repository to local machine

Practical:

- Create a Git repository using the cheat sheet commands
- Keep a word document named “Screenshots” open to put screenshots in
- Edit the “Student” file to include your Student Name & Student ID
- Commit changes
- Create IgnoreExample.txt
- Edit content of IgnoreExample.txt
- Use git status and screenshot the outcome
- Commit changes
- Add IgnoreExample.txt to .gitignore file
- Edit content of IgnoreExample.txt
- Use git status and screenshot the outcome
- Create a new branch using the git branch
- Switch to the new branch
- Edit BranchExample.txt (create this if not present) via the notepad command
  notepad BranchExample.txt
- Use git add to update branch example, commit and use the notepad command again
- Screenshot content
- Switch back to the master branch
- Use the same notepad command
- Screenshot the outcome
- Remove RemoveExample.txt (create one first if not present)
- Commit
- Use git status and screenshot outcome

- Use the clean command to remove the file completely
- Use git status and screenshot outcome
- Use git add to add your word document with the screenshots to the repository; it should look something like this:
  git add Screenshots.docx
- Commit

The submission deadline is the **end of the Lab**, late submissions will **not** be graded. If you have any issues or questions, contact the TA.