**Instructions for source control for git**

Create a new repository:

Create a new directory, then open up command prompt and navigate to that directory then type “git init” to create a new git repository.

Checkout a repository:

Create a working copy of a local repository by running the command:

git clone /path/to/repository

When using a **remote server**, your command will be:

git clone username@host:/path/to/repository

Workflow:

Your local repository consists of three “trees” maintained by git. The first one is your “Working Directory” which holds the actual files. The second one is the “Index” which acts as a staging area and finally the “Head” which points to the last commit you’ve made.

Add & Commit:

You can add changes using “git add <filename>” or “git add \*”

After adding the changes, to actually commit these changes use “git commit -m “Commit message”.

After this, the file is committed to the head, but not in the remote repository yet.

Pushing Changes:

Your changes are now in the “Head” of your local working copy. To send those changes to your remote repository, execute “git push origin master”. You can change “master” to whatever branch you want to push your changes to.

If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with “git remote add origin <server>”. Now you are able to push your changes to the selected remote server.

Branches:

Branches are used to develop features isolated from each other. The “master” branch is the “default” branch when you create a repository. Use other branches for development and merge them back to the master branch upon completion.

Create a new branch named “feature\_x” and switch to it using “git checkout -b feature\_x”.

You can switch back to master “git checkout master”.

You can delete the branch again using “git branch -d feature\_x”.

A branch is not available to others unless you push the branch to your remote repository by executing “git push origin <branch>”.

Update & merge:

To update your local repository to the newest commit, execute “git pull” in your working directory to fetch and merge remote changes.

To merge another branch into your active branch (e.g. master), use “git merge <branch>” in both cases git tries to auto-merge changes. Unfortunately, this is not always possible and results in “merge conflicts”.

You are responsible to merge those conflicts manually by editing the files shown by git. After changing, you need to mark them as merged with “git add <filename>” before merging changes, you can also preview them by using “git diff <source\_branch> <target\_branch>”.

OISAGAOSIGJASOIGJSAOGIJAROGIJRGOIRJGOAIRJGOIRJGORIGJORAIGJ