1 What is Git

1. Used to tracking changes, especilly text.
2. VCS: Version control system.
3. SCM: Source code management.
4. It is a distributed version control system:
   (a) Different users or teams of users maintain their own repository, instead of working from a central one.
   (b) Changes are stored as "changed sets" or "patches", this tracks changes and not versions.
   (c) No need to communicate with a central server.
   (d) Faster.
   (e) No network access required.
   (f) No single failure point.

2 Configuring Git

1. There are three places where git stores configuration information.
   (a) System: /etc/gitconfig
   (b) User: /.gitconfig
   (c) Project: my_project/.git/config
2. Commands for each:
   (a) git config –system
   (b) git config –global
   (c) git config
3. To dislpay configurations use git config –list OR u can specify which one.

3 Getting Started

1. git init //Initialize repository
2. To stop VC then remove .git file.
3. Usually only ever need to edit the config file in .git.
4. It contains the project configurations.
5. Basic git workflow:
   (a) make changes. //Add a new file
   (b) add the changes. //git add .

(c) commit changes to the repository with message. //git commit -m "Initial commit"

6. Write commit in the present tense.
7. git log
8. git log –help

4 Git Concepts and Architecture

1. Git uses the three-tree architecture.
2. This allows you to choose what you would like to commit.
3. The HEAD pointer is a reference to the most recent commint of the checkedout branch.
4. SHA is a unique 40 character hexadecimal string assigned to each commit.

5 Making Chnages to Files

1. git status //Shows difference between the working directory, staging index, and the repository.
2. git diff //Compare changes for repository and staging index between working directory.
3. git diff –staged //Compare changes between repository and staging index.
4. git rm file_name.
5. git mv file newName //Renaming or moving are the same.

6 Using Git with a Rreal Project

1. git commit -am "Initial commit" //Does all in one go, caveat, if u delete a file or it is not tracked this will not commit it.

7 Remotes

1. You push your repository to the remote server.
2. This create a new pointer to point at the latest commit, known as origin/master.
3. You need to pull changes from other people by using a fetch.
4. If your master is out of sync with the latest commint then need to merge.
5. git remote gives list of all remotes it knows about.
6. git remote add <name> <url>

7. Can have as many remotes as you would like, usually first one is known as origin.
8. git remote rm <name> //To remove remote.
9. git clone <url> [newName]
10. Fetch syncronizes origin/master with remote repository.(Need Internet).
11. git fetch //Need name if we have more than 1 repository.
12. fetch only updates origin/master, not master.
13. Tips:
    (a) fetch before you work.
    (b) fetch before you push.
    (c) fetch often.
    (d) origin/master is just a remote brach, though we can not check it out.