Git Contd

Merge Conflict

- Refer Here for merge conflict documentation
- Watch classroom video for examples

How git works

- As discussed we have two refs
  - HEAD: we have observed branches
  - TAGS
- Branch always points to latest commit done on it
- To understand git we need to understand the concept of Hashing
- Git uses SHA1
- We will be using two plumbing commands `git cat-file -t` and `git cat-file -p`
- Git commit is sha1 hash of
  - author
  - committer
  - commit message
  - previous commit (parent)
  - contents of commit
- In the world of git
  - tree represents folder
  - blob represents file

Detached Head

- This is a state where HEAD points directly to a commit or TAG
- In git, HEAD is supposed to be looking at branch
- When will this situation arise
  - git checkout `commit-id`
  - git checkout `tag`
- If we want to go back to a previous commit and make changes then the changes might become untracable, for that reason as discussed in these situations make changes and create a new branch on that change.
- Refer classroom video for making changes from DETACHED Head state