CS5500 - Managing Software Development

Git Cheat Sheet

SETUP

Set name associated with commits

```
git config --global user.name "[full name]"
```

Set email associated with commits

```
git config --global user.email "[valid email]"
```

CREATE

Clone an existing repository

```
git clone [url]
```

Create a local repository

```
git init
```

LOCAL CHANGES

Changed files in working directory

```
git status
```

Changes in tracked files that are not staged

```
git diff
```

Staged changes that are not committed

```
git diff --staged
```

Add all changes to the next commit

```
git add .
```

Add all changes in a file to the next commit

```
git add [file]
```

Add some changes in a file to next commit

```
git add -p [file]
```

Commit all changes in tracked files

```
git commit -a
```

Commit previously staged changes

```
git commit -m "[commit message]"
```

Change the last commit

```
git commit --amend
```

COMMIT HISTORY

Show all commits to active branch

```
git log
```

Show commits to a specific file

```
git log -p [file]
```

Line by line revision history in a file

```
git blame [file]
```

BRANCH & TAG

List all existing branches

```
git branch -av
```

Create a new branch

```
git branch [name]
```

Switch to another branch

```
git checkout [name]
```

Create a tracking branch

```
git checkout --track [name]
```

Delete a local branch

```
git branch -d [name]
```

Mark the current commit with a tag

```
git tag [tag name]
```

MERGE & REBASE

Merge a branch to current one

```
git merge [branch name]
```

Rebase current branch onto another branch

```
git rebase [branch name]
```

Abort a rebase

```
git rebase --abort
```

Continue rebase after resolving conflicts

```
git rebase --continue
```

Use configured tool to resolve conflicts

```
git mergetool
```

Use editor to manually resolve conflicts

```
git add [resolved file]
git rm [resolved file]
```

INSPECT & COMPARE

Show commits in branch 1 that are not in 2

```
git log [branch 2]..[branch 1]
```

Show diff of what is in branch 1 but not in 2

```
git diff [branch 2]...[branch 1]
```

Show changes to a file across file renames

```
git log --follow [file]
```

Show any object in human readable format

```
git show [sha]
```

UPDATE & PUBLISH

List all currently configured remotes

```
git remote -v
```

Show information about a remote

```
git remote show [name]
```

Add a new remote repository

```
git remote add [alias] [url]
```

Fetch all branches from remote

```
git fetch [alias]
```

Merge remote branch to active branch

```
git merge [alias]/[branch]
```

Fetch & merge changes from remote branch

```
git pull
```

Publish local changes to remote

```
git push [remote] [branch]
```

Delete a remote branch

```
git branch -dr [remote branch]
```

Publish tags

```
git push --tags
```

UNDO & REWRITE HISTORY

Discarl local changes in working directory

```
git reset --hard HEAD
```

Discard local changes in a file

```
git checkout HEAD [file]
```

Revert a commit

```
git revert [commit]
```

Reset HEAD to a previous commit and preserve changes as unstaged changes

```
git reset [commit]
```

Reset HEAD to a previous commit and discard all changes

```
git reset --hard [commit]
```

Reset HEAD to a previous commit and preserve uncommitted local changes

```
git reset --keep [commit]
```

Apply commits of current branch ahead of specified one

```
git rebase [branch]
```