Joseph Chung

Git Workflow

First time working on the project:

```
git clone git@csitgitlab.monmouth.edu:hawkscode/hawkscode.git
```

(SSH must be set up, see “ssh.pdf”)

```
cd hawkscode
git config --global user.name “YOUR_NAME”
git config –global user.email “YOUR_EMAIL”
```

Download a diff/merge tool. The simplest way I’ve seen to solve merge conflicts is with a text editor like VS Code: https://code.visualstudio.com/

```
git config --global merge.tool vscode
git config --global mergetool.vscode.cmd "code --wait $MERGED"
git config --global diff.tool vscode
git config --global difftool.vscode.cmd "code --wait --diff $LOCAL $REMOTE"
```

Creating a new branch:

```
git pull
```

Before creating a new branch, pull the changes from upstream. Your master needs to be up to date.

```
git checkout -b my-new-branch
```

This creates a new branch called my-new-branch and switches to this branch. Alternatively, you can also run `git branch my-new-branch` but you must switch to that branch using `git checkout my-new-branch`.

Pushing new changes:

```
git status
```

This will show all your changed/untracked files

```
git add .
```

This will add all your files to the staging area to be committed.

```
git commit -m “My message”
```

This commits all your staged changes and adds a message to the commit. Conventionally, commit messages are in present tense e.g. “Adding file.txt”, “Fixing bugs”, “Deleting file.txt”

```
git pull origin master
```

Get changes from remote master branch. MAY RESULT IN MERGE CONFLICTS. Resolve them then repeat the previous steps again before pushing.

```
git push -u origin my-new-branch
```

This pushes all your changes from your own branch to the remote Git repository, you only have to run this full command once and then you can run `git push` anytime thereafter.

Deleting a branch:

```
git checkout master
```

Switches to master branch because you can’t delete the branch that you’re currently on.

```
git push --delete origin my-new-branch
```

Deletes the remote branch.

```
git branch -d my-new-branch
```

Deletes the local branch.

```
git fetch -p
```

This will update any deleted branches, this should be run by everyone to reflect the branch deletion.

Merging:

```
git pull
git merge origin/my-new-feature
```

This will merge the remote master branch with a local branch.

Undo Merges:

```
git reflog
```

check which commit is the one prior the merge, copy the commit id (looks like random letters and numbers)

```
git reset --hard COMMIT_SHA
```

replace COMMIT_SHA with the id you just copied

Misc.:

```
git branch -a
```

This lists all branches, and the pointer(*) should be next to the branch you are on.