GIT (CONT.)

MERGETOOLS

- Resolve conflicts by visualizing files being merged
- Many different options: meld
- Must configure git first:

  ```
  git config --global merge.tool meld
  git config --global
  mergetool.keepBackup false
  ```

- After running `git merge` run `git mergetool`
- After resolving all in mergetool, add all files then commit

STASHING CHANGES

- Temporarily saving changes to files to use later
- `git stash` - takes staged and unstaged changes to tracked files and saves for later, removing these changes from your working directory.
- Use `-u` option to stash untracked
- `git stash pop` - get stashed changes back
- Also possible to access more than just most recent stash and add messages when stashing

REVERTING COMMITS

- `git revert <commit>`
- Examples:
  - `git revert HEAD~1`
  - `git revert SHA`
- Does not rewrite history - makes a new "revert" commit

VIEWING PREVIOUS VERSIONS OF FILES

- `git restore --source=<commit> filepath`
  - can replace with branchname
  - filepath can be `.` or use globbing (e.g. `*.c`)
- `git checkout <commit> filename`
  - DON'T do `git checkout <commit>` -> detached HEAD
- `git show <commit>:filename`