**Ch. 21 - Git and Github:**

Version control – Git allows us to save different versions of our program. We can rollback to a stable version if we made a change that breaks our code or continue to keep updating each version as we go.

**How to set up a local repository using git:**

git init

creates empty git repository in your current directory

git status

Shows you which files will be added to your staging area (intermediate area)

Non-committed files will be displayed as red

Committed files will be displayed in green

Modified files (from previous commit) will be displayed in red as well

git add fileName.extension

Adds a file to your staging area to be committed.

git add .

Adds all untracked files present in current directory (not previously committed)

git rm --cached -r .

Removes all files from the staging area.

git commit -m “message goes here”

Commits the files in the staging area as their own version

Write in present tense

git log

Displays a log of your commits, author, commit messages and time of commit.

Also displays unique commit hash which identifies each commit.

git checkout fileName.extension

Rolls back to the last version in our git repository

git diff fileName.extension

Shows you the difference between local file and last committed version

**How to ignore certain files from git:**

This is used for sensitive information (passwords/apiKeys) or user settings/preferences (DS\_Store)

Create your .gitignore file in your directory using your CLI

.touch .gitignore

Add the file names in your .gitignore file you do not want to commit

#Comment here

Allows you to put comments in your .gitignore file

\*.extension

This ignores all files with the specified extension from being committed

Github.com/github/gitignore

Contains many prebuilt .gitignore files you can use for your projects

**How to set up an external repository using github (git push):**

Set up your repository on github.com

In your CLI, type:

git remote add origin remoteRepositoryURL

This transfers your local repository to a remote repository you have on github

The name of your remote repository is usually kept as ‘origin’ (best practice)

git push -u origin master

pushes local repository to remote repository titled ‘origin’ on the master branch using the -u option.

Master branch is default/main branch of your repository

Master branch is your main branch of commits/save points. It is sequential. Main progress is saved here.

You can have a local repository in parallel with a remote repository. By synchronizing both files, you can keep both up to date.

**How to clone a remote repository onto a local repository (git clone):**

git clone remoteRepositoryURL

Clones the remote repository onto your local machine in the directory specified.

**Side branches:**

Side branches allow for multiple branches of the project to be worked on simultaneously. A side branch may be used to work on new features, bug fixes, experimental ideas, etc. You can make multiple side branches and you can also merge a side branch back into the main branch.

git branch branchName

Creates a side branch with name branchName

git branch

displays all current branches of your repository

The current repository is identified with an \*

git checkout branchName

Switches from current branch to side branch named branchName

**Forking (copying a remote repository onto your own remote repository):**

Once changes have been made on a forked remote repository, a pull request can be made from the original owner to the forked repository owner to incorporate/merge those changes onto the original remote repository.

\*\*\*

Git fetch updates your local repository with your remote repository

Git Merge merges your head branch with your remote branch/most up to date branch.

Git Pull fetches and merges your head branch all in one.