Questions 1-9 cover sections 1.1 to 1.3 of the ProGit assigned reading.

1. Version Control is a system that records \_\_\_\_\_\_\_\_\_ to a file or set of files over time so that you can recall \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ later.
2. Why is simply making a copy of your files an ineffective version control approach?
3. Version Control systems foster \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ between developers.
4. What is the main disadvantage of a centralized VCS?
5. What is the main advantage of a distributed VCS?
6. How long has the Git VCS been around?
7. Git does not use simple serial numbers to track successive versions of a file. What does it use instead? Why?
8. List the 3 states that a file can reside in when it is being maintained by Git:
9. List the 3 main sections of a Git project, and briefly state what each is used for:

Questions 10-16 cover sections 2.1, 2.2 and 2.4 of the ProGit assigned reading.

1. What are the two main approaches to creating a Git project?
2. \_\_\_\_\_\_\_\_\_\_ files are those that are not managed by Git – they are neither committed/unmodified, modified, nor staged.
3. The main “branch” of the files maintained by Git is typically called \_\_\_\_\_\_\_\_\_\_\_\_\_\_.
4. When a file is untracked, what effect does the “git add” command have on the file?
5. When a file is tracked, what effect does the “git add” command have on the file?
6. What is the effect of applying the “git commit” command to a staged file?
7. Say you have modified a tracked file in your working directory, but want to disregard the modifications and recover the previous version. What command is used to do this?

Questions 17-19 cover section 2.5 of the ProGit assigned reading.

1. \_\_\_\_\_\_\_\_\_\_\_ repositories are versions of your Git project that are hosted on the internet or network.
2. Explain the difference between Fetching and Pulling from a Remote?
3. What is the purpose of Pushing to a Remote?