**What happens when you initialize a repository? Why do you need to do it?**

**Git starts to track our file history when we initialize it as a repository. We need to do it in order to view our file history and to use version-control on it.**

**How is the staging area different from the working directory and the repository? What value do you think it offers?**

**The staging area allows us to stage multiple files intermediate of the commit and working directory. It helps us in staging multiple files before making a commit.**

**How can you use the staging area to make sure you have one commit per logical change?**

**By using the staging area we can compare the staging area and the repository to ensure that there is one commit per logical change.**

**What are some situations when branches would be helpful in keeping your history organized? How would branches help?**

**When we are not sure about what we are doing we will include it in the final version or when we are creating another type of file in it which is different from the master. They help us in working on files without affecting the master branch.**

**How do the diagrams help you visualize the branch structure?**

**The branches help us in visualising the flow of branches from child to parent and in determining that which branches are retrievable.**

**What are the pros and cons of Git’s automatic merging vs. always doing merges manually?**

**The pros of using automatic merging is that it saves us a lot of time and effort required in making manual commits whereas manual commits help us in resolving the merge conflicts.**