1. **What is Git?**
   Ans. A git is a version control system. It is a repository where we can store Our code, update our code and if we are working in a team then if we push Our code on git then everyone in our team can access that code and update The code and it will keep the record of all those things.
   Now suppose if there is dependency of the code i.e we need to work with other developers then to collaborate with that developer we need the help of Git. Also If we need to integrate different code together then in that case also we will take help from git.
   . Git is used for tracking code changes
   . Git is used for tracking who made changes like history of the files
   . Git is used for coding collaboration

2. **What do you understand by the term 'version control system'?**
   Ans. Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done in the code.
   The version control system is a collection of software tools that help a team to manage changes in a source code. It uses a special kind of database to keep track of every modification to the code.

3. **What is GitHub?**
   Ans. GitHub is a Git repository hosting service. GitHub also facilitates with many of its features, such as access control and collaboration. It provides a Web-based graphical interface. It hosts source code of your project in the form of different programming languages and keeps track of the various changes made by programmers.
   Features of Github are :- Collaboration

- Integrated issue and bug tracking
- Graphical representation of branches
- Git repositories hosting
- Project management
- Team management
- Code hosting
- Track and assign tasks

4. **Mention some popular Git hosting services?**

   Ans.

- **Bitbucket.**
- **GitLab.**
- **Perforce.**
- **Beanstalk.**
- **Amazon AWS CodeCommit.**
- **Codebase.**
- **Microsoft Azure DevOps.**
- **SourceForge.**

5. **Different types of version control system?**

   Ans. **There are two types of version control system**

   .Centralized version control system :- In Centralized version control system there will be one main server or Repository, and all of those who are associated with that project have access of that main server. And Whatever the developer is coding will have to push or commit it to main server and once you had committed that code then everyone who are working on that project can access and update that code.

   In centralized version control system you can collaborate with your co-workers and get the work done.

   .Distributive version control system :- Just like centralized version control system in distributive version control system also there will be one main Repository or server where all the works will be stored but in this there will be one local Repository or local server present with every developer so that they can firstly store his work in his local server and then from local server he will push those codes to the main server.

6. **What benefits come with using GIT?**

   Ans. **One of the biggest advantages of Git is its branching capabilities. Unlike centralized version control systems, Git branches are cheap and easy to merge. This facilitates the feature branch workflow popular with many Git users. Feature branches provide an isolated environment for every change to your codebase.**

7. **What is a git repository?**

   Ans. A Git repository stores the history of changes made to your codebase by a collection of files and folders. As a developer, I have found this to be extremely useful since it allows me to keep a single view of the project codebase, back up a backup copy of the entire project history, easily retrieve older versions of the whole

codebase or individual files, debug code, find out who wrote a specific change and a lot more. This is extremely useful for development teams.

8. How can you initialize a repository in Git?

   Ans. To create a new repo, you'll use the `git init` command. `git init` is a one-time command you use during the initial setup of a new repo. Executing this command will create a new `.git` subdirectory in your current working directory. This will also create a new main branch.