1.Ans). Version control is also known as source control or revision control is an important software development practice for tracking and managing changes made to code and other files. It is closely related to source code management. A version control system (VCS) tracks changes to a file or set of files over time.

2.Ans). Version control systems allow multiple developers, designers, and team members to work together on the same project. It helps them work smarter and faster! A version control system is critical to ensure everyone has access to the latest code and modifications are tracked.

3.Ans). The various types of the version control systems are:  
 1. Local Version Control System  
 2. Centralized Version Control System  
 3. Distributed Version Control System

### 1.Local Version Control System: Local version control system maintains track of files within the local system. This approach is very common and simple. This type is also error prone which means the chances of accidentally writing to the wrong file is higher.

### 2. Centralized Version Control Systems:- In this approach, all the changes in the files are tracked under the centralized server. The centralized server includes all the information of versioned files, and list of clients that check out files from that central place. **Example: Tortoise SVN**

### 3. Distributed Version Control System:Distributed version control systems come into picture to overcome the drawback of centralized version control system. The clients completely clone the repository including its full history. If any server dies, any of the client repositories can be copied on to the server which help restore the server.Every clone is considered as a full backup of all the data. **Example: Git**

### **4.Ans). Centralized version control is easier to learn than distributed. If you are a beginner you’ll have to remember all the commands for all the operations in DVCS and working on DVCS might be confusing initially. CVCS is easy to learn and easy to set up.**

### **DVCS has the biggest advantage in that it allows you to work offline and gives flexibility. You have the entire history of the code in your own hard drive, so all the changes you will be making in your own server or to your own repository which doesn’t require an internet connection, but this is not in the case of CVCS.**

### **DVCS is faster than CVCS because you don’t need to communicate with the remote server for each and every command. You do everything locally which gives you the benefit to work faster than CVCS.**

### **Working on branches is easy in DVCS. Every developer has an entire history of the code in DVCS, so developers can share their changes before merging all the ‘sets of changes to the remote server. In CVCS it’s difficult and time-consuming to work on branches because it requires to communicate with the server directly.**

### **If the project has a long history or the project contain large binary files, in that case, downloading the entire project in DVCS can take more time and space than usual, whereas in CVCS you just need to get few lines of code because you don’t need to save the entire history or complete project in your own server so there is no requirement for additional space.**

### **If the main server goes down or it crashes in DVCS, you can still get the backup or entire history of the code from your local repository or server where the full revision of the code is already saved. This is not in the case of CVCS, there is just a single remote server that has entire code history.**

### **Merge conflicts with other developer’s code are less in DVCS. Because every developer work on their own piece of code. Merge conflicts are more in CVCS in comparison to DVCS.**

### **In DVCS, sometimes developers take the advantage of having the entire history of the code and they may work for too long in isolation which is not a good thing. This is not in the case of CVCS.**

### **5.Ans). Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development.**

### **6.Ans).**

### **Works on a distributed system: A distributed system is one that allows collaborators to access the central repository using a VCS, even from the remotest corner of the world. As Git maintains a snapshot every time a user pulls a file, the risk of data loss due to system failure or lack of Internet connection is mitigated. Users are allowed to work on the same bit of code simultaneously without getting interfered by others.**

### **Compatible with all operating systems: Git is compatible with almost all operating systems that are available today. Even the repositories created by other version control systems can be accessed by the Git repository.**

### **Allows for non-linear development: As users from remote parts of the world can access the Git repository, work on it, and update the project at any time they want, Git allows for development in a non-linear fashion. Git supports such a kind of development by providing its branching and merging features, and it uses specific tools for navigating through them. The projects are viewed in a tree form.**

### **Branches like a tree: While users are working on their projects, branches parallel to the main project file are created, so the original code is not affected. There is no restriction upon the number of branches created.**

### **Light as a cotton ball: One might think that making multiple copies of data from a central repository to a local one will eventually lead the system to crash due to overload. But, Git has got it covered. It compresses the data in such a way that it takes up minimal space, and whenever you need to retrieve data, the reverse technique is used. This helps save a lot of memory.**

### **Fast as a flash: Unlike other version control systems, Git is written in a language known to be the closest to the machine language, that is, C. Hence, it processes information much faster.**

### **Reliable: There will never be an issue of data loss as long as the copies of data in the central repositories are available in the local repositories of different collaborators.**

**7.Ans).**

**Creating a new branch -** git branch <branch-name>

**Git clone** - git clone <https://name-of-the-repository-link>

**Git to add everything at once** - git add -A

8.Ans). Git is a tool that’s used to manage multiple versions of source code edits that are then transferred to files in a Git repository, GitHub serves as a location for uploading copies of a Git repository.In a sense, then, there’s no comparison when it comes to Git vs. GitHub as far as their function.

9.Ans). Check your version by this command:- git –version or git -v.

10.Ans). git add -A

12.Ans). Initializing a new repository: git init

15.Ans). TO Git commit:-

git add git commit git diff git stash .gitignore

16. git commit -m “New email”