**Q. What is GitHub?**

**A.**

* GitHub is a web-based Git repository hosting service.
* GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.
* It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.
* As with most other distributed version control systems, and unlike most client–server systems, every Git directory on every computer is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server.
* In general, GitHub is a combination of, “A Publishing tool, Version Control System, Collaboration Platform”.

**Q. When was it created?**

**A.**

* Development of the GitHub platform began on October 1, 2007.
* It had been made available for a few months prior as a beta release.
* It was finally launched on April 10, 2008.

**Q. Why?**

**A.** GitHub has many advantages, such as,

1. Versioning Project and having Version Control
   * Version Control System (VCS) is used for giving versions to the different release of the projects. VCS also maintains the past history of the code. The codes will be saved and we can look back to check the history of the code. So it’s good for keeping the code clean and well maintained.
2. Publishing projects – After publishing the project, the user can download it, install it, and work with it, and even customize it if anyone wants to.
3. Collaborating projects – If some group of developers are working on a project, even if they are not local, say they are all over the world; they can use it as a sort of a hub and contribute their work, from any place.

**Q. By who?**

**A.** The founders of GitHub are Tom Preston-Werner, Chris Wanstrath, PJ Hyett.

**Q. What similar platforms exist?**

**A.** Similar Platforms are,

* GitLab
* [Bitbucket](http://alternativeto.net/software/bitbucket/)
* SourceForge
* LaunchPad
* Trac
* GitBucket
* CodePlex
* Assembla
* RhodeCode
* CodeBase

Define the following terms in the context of Git (2 lines maximum):

* **Repository**
  + A directory where Git has been initialized to start version controlling your files.
  + A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and anything the project needs.
* **Commit**
  + On GitHub, saved changes are called commits. Each commit has an associated commit message, which is a description explaining why a particular change was made.
  + Commit messages capture the history of your changes, so other contributors can understand what you’ve done and why.
* **Push**
  + Pushing refers to sending our committed changes to a remote repository such as GitHub.com. For instance, if we change something locally, we'd want to then push those changes so that others may access them.
* **Branch**
  + Branching is the way to work on different versions of a repository at one time.
  + When we create a branch off the master branch, we’re making a copy, or snapshot, of master as it was at that point in time. If someone else made changes to the master branch while we are working on our branch, we could pull in those updates.
* **Fork**
  + A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original.
* **Merge**
  + Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another. This often happens as a Pull Request (which can be thought of as a request to merge), or via the command line.
* **Clone**
  + A clone is a copy of a repository that lives on our computer instead of on a website's server somewhere, or the act of making that copy.
  + With our clone we can edit the files in our preferred editor and use Git to keep track of our changes without having us to be online.
* **Pull**
  + Pull refers to when we are fetching in changes and merging them. For instance, if someone has edited the remote file we're both working on, we'll want to pull in those changes to our local copy so that it's up to date.
* **Pull request** 
  + Pull requests let us tell others about changes we've pushed to a repository on GitHub.
  + Once a pull request is opened, we can discuss and review the potential changes with collaborators and add follow-up commits before the changes are merged into the repository.