**Java Week 2**  
  
**Git Tools and Packages:**

1. **Git:** Git is a distributed version control system used for tracking changes in source code during software development. It allows multiple developers to collaborate on projects and keeps track of changes made to the codebase.
2. **GitHub:** GitHub is a web-based platform for hosting Git repositories. It provides collaboration features like pull requests, issue tracking, and project management tools.
3. **GitLab:** Like GitHub, GitLab is a web-based Git repository manager providing CI/CD (Continuous Integration/Continuous Deployment) pipelines, issue tracking, and other DevOps functionalities.
4. **Bitbucket:** Bitbucket is another web-based platform that offers Git repository management along with collaboration features, especially popular among teams using the Atlassian suite of tools.
5. **SourceTree:** SourceTree is a free GUI Git client that simplifies how you interact with your repositories. It provides an easy-to-use interface for managing branches, commits, and merges.
6. **GitKraken:** GitKraken is another popular GUI Git client known for its user-friendly interface and features like Gitflow integration, drag-and-drop functionality, and visualization tools.

**Git Basic Commands:**

1. **git init:** Initializes a new Git repository in the current directory.
2. **git clone:** Clones an existing repository into a new directory.
3. **git add:** Adds changes in the working directory to the staging area.
4. **git commit** Records changes in the staging area to the repository.
5. **git status:** Displays the status of the working directory and staging area.
6. **git push:** Uploads local repository changes to a remote repository.
7. **git pull:** Fetches changes from a remote repository and merges them into the local branch.
8. **git branch:** Lists, creates, or deletes branches within the repository.
9. **git merge:** Combines changes from different branches into the current branch.
10. **git checkout:** Switches branches or restores working tree files.

**Git Log and Git Checkout:**

1. **git log:** Shows a history of commits in the repository. It displays commit hashes, authors, dates, and commit messages.
2. **git log --oneline:** A concise way to view commit history, showing abbreviated commit hashes and commit messages on a single line.
3. **git checkout:** Used to switch branches or restore files in the working directory to a previous state.

For example:

* **git checkout <branch\_name>** switches to the specified branch.
* **git checkout <commit\_hash> <file\_path>** restores a specific file from a particular commit.

These tools, commands, and actions in Git are fundamental for version control, collaboration, and managing changes in software development projects.