DVCS & Version Control Study Guide

Topic 1: DVCS (Distributed Version Control System)

Definition:
A Distributed Version Control System (DVCS) is a system where every user has a full copy of the repository, including its history. Changes can be made locally and later synchronized with others.

Key Features:
- Local repository: Users can commit, branch, and view history without internet access.
- Distributed nature: Every clone is a full backup of the repository.
- Collaboration: Changes can be shared between repositories using push/pull operations.
- History tracking: All commits and changes are tracked independently on each local repository.

Advantages over Centralized VCS (CVCS):
Repository: Local full copy vs only central server
Offline work: Yes vs No
Backup: Yes vs No
Merging: Easy vs Hard

Popular DVCS Examples:
- Git
- Mercurial
- Bazaar

Common DVCS Workflow:
1. Clone a repository
2. Make changes locally
3. Commit changes locally
4. Push changes to remote
5. Pull changes from others

Visual Summary (Flow):
Local Repo -> Commit -> Push -> Remote Repo -> Pull -> Local Repo

Topic 2: Key Characters

Definition:

Key Characters refer to important people or roles who influence, manage, or contribute to the project.

Examples of Key Characters:
- Maintainer: Manages the repository, reviews pull requests, merges changes.
- Contributor: Adds code, documentation, or other improvements.
- Committer: Has permission to commit directly.
- Reviewer: Reviews code changes, suggests improvements, and approves merges.

Importance:
- Ensures project quality and consistency.
- Tracks accountability.
- Facilitates collaboration.

Roles Summary Table:
Maintainer -> Oversees repo, merges PRs
Contributor -> Submits changes, fixes bugs
Committer -> Commits code directly
Reviewer -> Reviews PRs, ensures code quality

Visual Summary:
Contributor -> Pull Request -> Reviewer -> Merge -> Maintainer

Topic 3: Comparison of Centralized VCS (CVCS) vs DVCS

Definition:
CVCS: Single central repository; all users commit directly to it.
DVCS: Each user has a full copy of the repository; allows local commits and offline work.

Key Differences:
Repository: Single central vs Local + remote
Offline Work: No vs Yes
Backup: No vs Yes
Branching & Merging: Hard vs Easy
Collaboration: Direct commit vs Pull/Push workflow
Examples: SVN, CVS vs Git, Mercurial, Bazaar

Advantages of DVCS over CVCS:

- No single point of failure.
- Offline commits.
- Easier branching & merging.
- Better support for distributed teams.

Visual Summary:

CVCS: User -> Commit -> Central Repo <- Commit <- User

DVCS: Local Repo -> Commit -> Push -> Remote Repo <- Pull <- Local Repo