**1. How Mocking Enhances Test-Driven Development (TDD)**

* Enables writing tests before the actual implementation (core idea of TDD).
* Allows isolation of the unit under test from external dependencies.
* Uses test doubles:
  + **Mock** – verifies interactions.
  + **Stub** – returns fixed data.
  + **Fake** – simplified working implementation.
* Improves test reliability, speed, and repeatability.
* Encourages better code design and modularity.

**2. Meaning and Purpose of Mocking in Unit Testing**

* Simulates real dependencies (e.g., database, API) during testing.
* Ensures tests focus only on the logic of the unit being tested.
* Allows control over return values and simulated failures.
* Verifies that specific methods were called with expected arguments.
* Makes tests faster, more stable, and independent of environment setup.

**3. Basics of Dependency Injection (DI) and Its Role in Unit Testing**

* **Dependency Injection** supplies objects with their dependencies externally.
* Promotes loose coupling and better testability.
* **Constructor Injection**: Dependencies passed via constructor.
* **Method Injection**: Dependencies passed through method parameters.
* Enables substitution with mocks during testing.
* Helps isolate the class being tested and improves maintainability.