1. What is TDD (Test-Driven Development)?

Test-Driven Development (TDD) is a software development methodology in which **test cases are written before the actual application code**. Development follows a short, repetitive cycle:

1. **Write a test** for a small functionality (test fails initially).
2. **Write the minimum code** required to pass the test.
3. **Refactor** the code while ensuring all tests continue to pass.

2. Advantages of TDD

1. **Early Bug Detection**
   Defects are identified at the unit level before integration.
2. **Improved Code Quality**
   Encourages modular, loosely coupled, and maintainable code.
3. **Better Requirement Understanding**
   Writing tests first forces developers to clearly understand requirements.
4. **Safe Refactoring**
   Existing tests act as a safety net during code changes.
5. **Higher Test Coverage**
   Code is naturally covered by unit tests from the beginning.
6. **Faster Debugging**
   Smaller code changes make identifying issues easier.

3. Disadvantages of TDD

1. **Initial Development Time Is Higher**
   Writing tests before code increases early development effort.
2. **Learning Curve**
   Requires solid understanding of testing frameworks and design principles.
3. **Not Ideal for UI-Heavy or Exploratory Work**
   Difficult to write tests upfront for rapidly changing user interfaces.
4. **Over-Testing Risk**
   Excessive or poorly designed tests can slow down maintenance.
5. **Maintenance Overhead**
   Test cases must be updated when requirements change.

4. Tools Used for TDD

Common tools used to implement TDD include:

a. Unit Testing Frameworks

- **JUnit** (Java)
- **TestNG** (Java)
- **NUnit** (.NET)
- **pytest / unittest** (Python)

b. Mocking Frameworks

- **Mockito**
- **EasyMock**
- **PowerMock**

c. Build & Automation Tools

- **Maven**
- **Gradle**
- **Ant**

d. CI/CD Tools (to enforce TDD)

- **Jenkins**
- **GitHub Actions**
- **GitLab CI**

5. Use of Property File – Why Not Read from Excel?

Purpose of a Property File

A **property file** (`.properties`) is used to store **configuration data** in key–value format, such as:

- URLs
- Database credentials
- Environment variables
- Timeout values

Example:

```
url=https://example.com
browser=chrome
timeout=30
```

Why Property Files Are Preferred Over Excel

Aspect	Property File	Excel File
Read Performance	Very fast	Slower
Dependencies	No external library	Requires Apache POI
Complexity	Simple key–value	Complex parsing
Usage	Configuration	Test data
CI/CD Friendly	Yes	Not ideal
Version Control	Easy diff	Binary / harder

When Excel Is Used Instead

Excel is preferred **only for test data-driven testing**, such as:

- Multiple input combinations
- Large datasets
- Business scenarios