**Day-3 (Assignment-1)**

**Q.1)** Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.

**TDD**:- Test-driven development (TDD) is a method of coding in which you first write a test and it fails, then write the code to pass the test of development, and clean up the code. This process recycle for one new feature or change. In other methods in which you write either all the code or all the tests first, TDD will combine and write tests and code together into one.

# Process of Test Driven Development (TDD)

It is the process in which test cases are written before the code that validates those cases. It depends on the repetition of a concise development cycle. Test-driven Development is a technique in which automated Unit tests are used to drive the design and free decoupling of dependencies.

The following sequence of steps is generally followed:

## Test-Driven Development (TDD) Infographi

**1) Write Test Cases First**

-> Begin by writing failing unit tests based on requirements.

**2) Code Implementation**

-> Develop the minimum code required to pass the tests.

**3) Run Tests**

-> Execute all tests to verify if the new code passes.

**4) Refactor Code**

-> Improve code structure without changing its behavior.

**5) Repeat**

-> Continuously add new tests and refactor existing code

## Three Phases of Test Driven Development ( TDD )

**1) Create precise tests**

Developers need to create exact unit tests to verify the functionality of specific features. They must ensure that the test compiles so that it can execute. In most cases, the test is bound to fail. This is a meaningful failure as developers create compact tests based on their assumptions of how the feature will behave.

**2) Correcting the Code**

Once a test fails, developers must make the minimal changes required to update the code to run successfully when re-executed.

**3) Refactor the Code**

Once the test runs successfully, check for redundancy or any possible code optimizations to enhance overall performance. Ensure that refactoring does not affect the external behavior of the program.

# Benefits of TDD

1. **Bug Reduction :** Early detection and correction of issues.
2. **Improved Design :** Encourages modular and cleaner code.
3. **Software Reliability :** Ensures each unit performs as expected.
4. **Efficiency :** Reduces time spent on debugging and rework.

# Test-Driven Development (TDD) fosters software reliability by systematically integrating testing into the development process from the outset. By writing tests before writing functional code, TDD ensures:

* **Early Detection of Issues:** Bugs and errors are identified immediately as tests are run continuously throughout development.
* **Improved Code Quality:** Encourages developers to write modular, well-structured code that adheres closely to requirements.
* **Maintainability:** Facilitates easier maintenance and refactoring as changes can be made confidently with tests ensuring existing functionality remains intact.
* **Documentation:** Tests serve as living documentation, providing insights into expected behavior and edge cases.