Assignment-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.

# **Test- Driven Development(TDD)**:

Test-driven development (TDD) is a software development approach where tests are written before the code implementation. In traditional software development, code is typically written first, followed by testing to ensure that it functions correctly. However, in TDD, the process is reversed: developers first write a failing test that describes a small unit of functionality, then write the code to pass that test, and finally refactor the code to improve its structure while ensuring that all tests still pass.

# **Write Test Cases**:

* Develop tests that define the desired behavior of the software. These tests are written before any code is implemented.

Run Tests:

* Execute the tests to ensure they fail. This step verifies that the tests are correctly assessing the desired behavior.

Write Code:

* Implement the minimum code required to pass the failing tests. This code should fulfill the requirements outlined by the tests.

Run Tests Again:

* Rerun the tests to validate the newly implemented code. Tests should now pass since the code has been designed to meet their criteria.

Refactor Code:

* Optimize the code structure without altering its functionality. Refactoring improves code readability, maintainability, and efficiency.

Repeat Steps:

* Iterate the process by writing new tests, running them, writing code to pass the tests, and refactoring as necessary.

## Benefits of TDD:

Reduced Bugs:

* By writing tests before code, developers catch bugs early in the development process, leading to fewer defects in the final product.

Improved Reliability:

* TDD ensures that software functions as intended by continuously testing and refining code to match desired behavior.

Enhanced Code Quality:

* The iterative nature of TDD encourages cleaner, modular code design, resulting in higher-quality software that is easier to maintain and extend.

Faster Development:

* Although initially perceived as time-consuming, TDD ultimately speeds up development by reducing debugging time and minimizing the need for extensive rework.