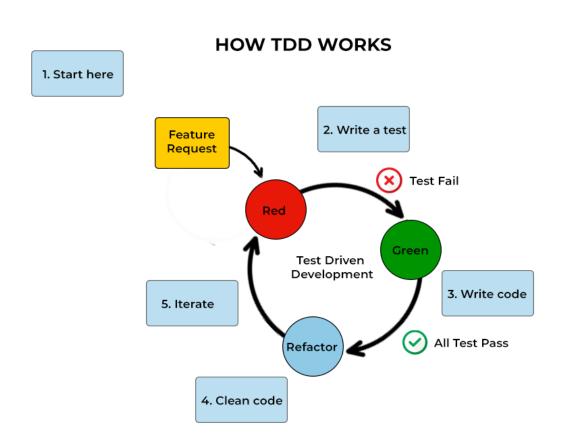**Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.**

Infographic: Test-Driven Development (TDD) Process

HOW TDD WORKS

1. Start here

Feature Request

2. Write a test

❌ Test Fail

Red

Green

Test Driven Development

3. Write code

5. Iterate

Refactor

✅ All Test Pass

4. Clean code

1.Introduction to TDD:

- Brief overview of TDD and its significance in software development.

2.Steps of TDD:

a. Write a Test

- Visual representation of a developer writing a test case before writing any production code.

b. Run the Test

- Illustration of running the test to see it fail initially, indicating that the test is effective.

<u>c. Write the Code</u>

- Showing the developer writing the minimum amount of code necessary to pass the test.

<u>d. Run All Tests</u>

- Showing all tests running, including the newly written one, to ensure nothing else has broken.

<u>e. Refactor Code</u>

- Representation of the developer refactoring the code while keeping all tests passing.

3.Benefits of TDD:

<u>1.Bug Reduction:</u>

- Visual comparison of bug occurrence rates between projects using TDD and those without.

<u>2.Improved Code Quality:</u>

- Representation of cleaner, more maintainable code resulting from TDD.

<u>Faster Development:</u>

- Illustration of how TDD can lead to faster development cycles due to fewer bugs and easier debugging.

<u>Increased Confidence:</u>

- Showing a developer with a confident stance, representing the assurance TDD provides in software reliability.

4.How TDD Fosters Software Reliability:

<u>1.Early Detection of Bugs</u>

- Visual of bugs being caught early in the development process, reducing their impact.

<u>2.Continuous Validation</u>

- Representation of continuous validation through automated tests, ensuring reliability at each stage.

<u>3.Incremental Development</u>

- Showing how TDD encourages incremental development, leading to more reliable and robust software over time.