Assignment 1:

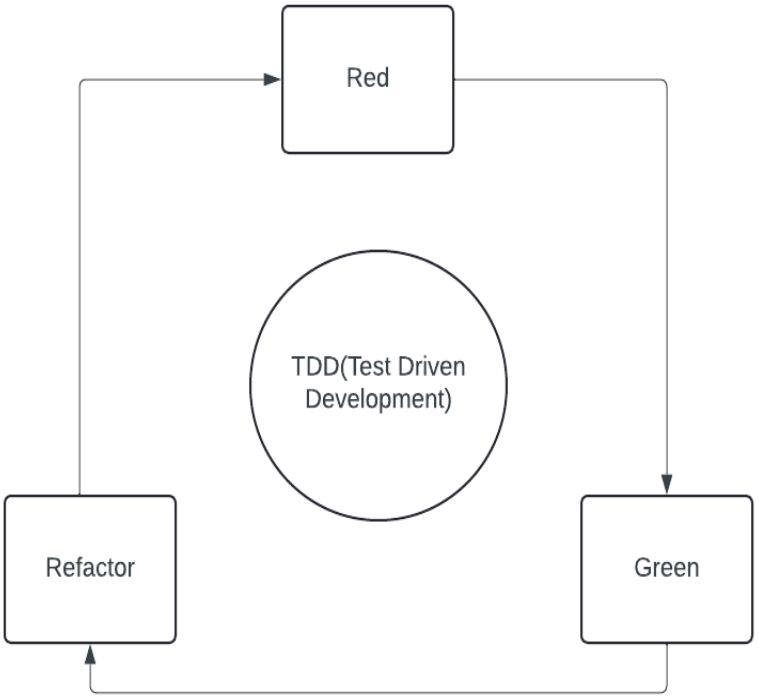
Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.

Ans:

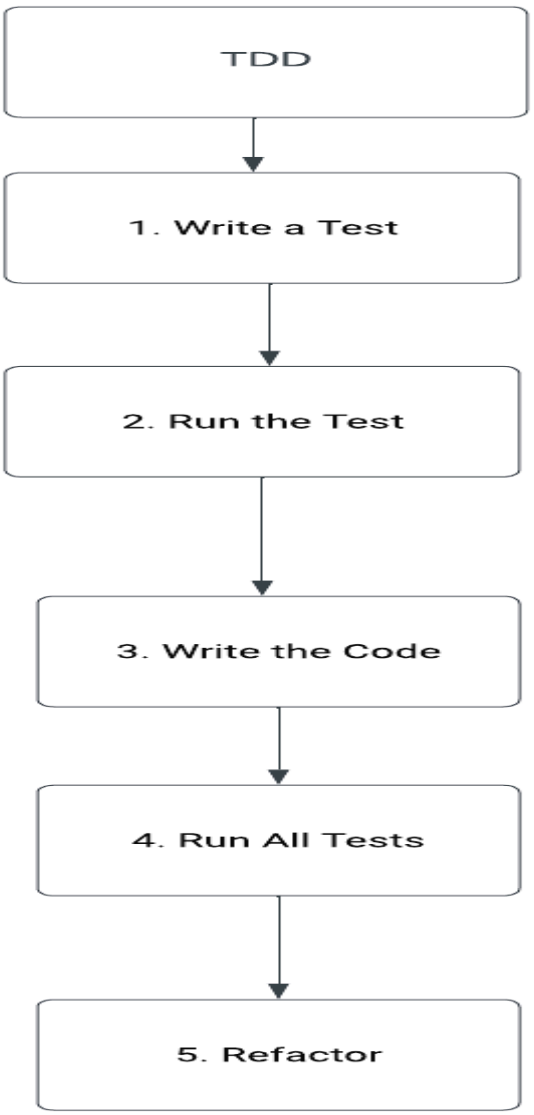
Title: Test-Driven Development (TDD) Process

Introduction:

- Test-Driven Development (TDD) is a software development approach where tests are written before the code.
- It follows a cycle of writing tests, writing code to pass those tests, and then refactoring.



1. **Red** – Create a test case and make it fail
2. **Green** – Make the test case pass by any means.
3. **Refactor** – Change the code to remove duplicate/redundancy.



1. Write a Test:

- Begin by writing a test that describes the desired behaviour of the code.
- Tests are typically written using testing frameworks like JUnit or Jasmine.
- Example: "Write a test to check if the login function returns true for valid credentials."

2. Run the Test:

- Execute the test to ensure it fails initially.
- This confirms that the test is valid and the feature doesn't exist yet.
- Example: "The test fails because the login function hasn't been implemented."

3. Write the Code:

- Implement the code necessary to pass the test.
- Focus only on making the test pass, not on writing perfect code.
- Example: "Write the login function to validate user credentials."

4. Run All Tests:

- Run all tests to ensure the new code hasn't broken any existing functionality.
- This step helps maintain the integrity of the codebase.
- Example: "Run all tests to verify that the new login function works and doesn't break any other features."

5. Refactor:

- Once the test passes, refactor the code to improve its design and maintainability.
- This step ensures the code remains clean and easy to understand.
- Example: "Refactor the login function to remove duplication and improve readability."

Benefits of TDD:

- Bug Reduction: By writing tests first, bugs are caught early in the development process, reducing the cost of fixing them later.
- Improved Software Reliability: TDD leads to more reliable software as it encourages writing small, focused units of code with clear requirements.

Conclusion:

- Test-Driven Development (TDD) promotes a disciplined approach to software development, resulting in higher quality code with fewer defects.