

## Infographic: Test-Driven Development (TDD) Process

### (i) Write Test Cases:-

Define the requirements and functionalities.

Write test cases for each functionality or unit of code.

### (ii) Run Tests:-

Execute the tests to ensure they fail. This ensures that the tests are valid and that the code to be written doesn't already satisfy them.

### (iii) Write Code:-

Develop the minimum code necessary to pass the failing tests.

Focus on writing only what is needed to make the tests pass.

### (iv) Run Tests Again:-

Execute the tests again to ensure the new code passes all tests.

If any tests fail, refine the code until all tests pass.

### (v) Refactor Code:-

Improve the code without changing its functionality.

Eliminate duplication, improve readability, and enhance performance.

### (vi) Repeat:-

Continue the cycle by adding new tests for additional functionalities.

Refactor existing code as necessary.

## Benefits of TDD:-->

**Bug Reduction:-** TDD helps identify and fix bugs early in the development process, reducing the likelihood of bugs in the final product.

**Improved Software Reliability:-** By continuously testing and refining code, TDD ensures that the software meets the specified requirements and functions reliably.

**Increased Developer Confidence:-** Developers gain confidence in their code as it is continuously validated through automated tests.

**Enhanced Code Maintainability:-** The focus on writing modular and testable code makes it easier to maintain and extend the software in the future.

### Conclusion:-->

Test-Driven Development (TDD) is a proven approach to software development that promotes code quality, reliability, and maintainability. By following a systematic process of writing tests before code, developers can create robust software that meets customer requirements while reducing bugs and enhancing overall product quality.

