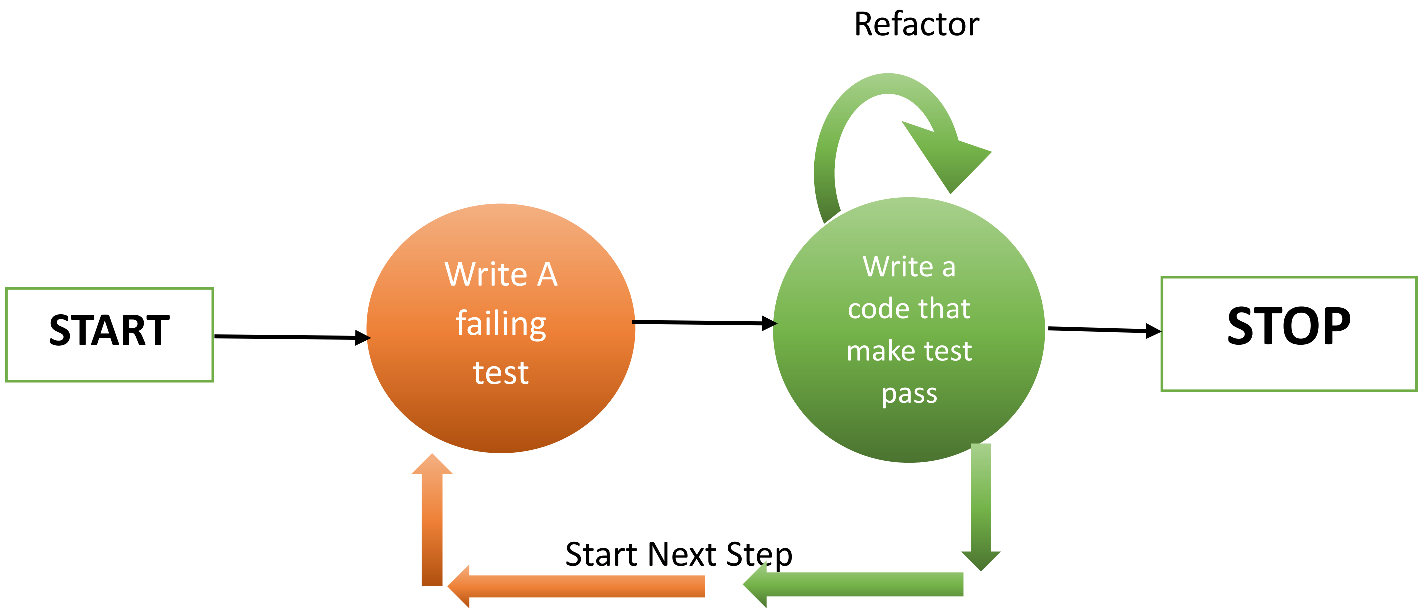
Q) create an infographic illustrating the test-driven development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction and how it fosters software reliability.

Sol) **Definition: -** The Test-Driven Development (TDD), also called Test Driven Design. It is the method of implementing the software programming that interlaces unit testing, programming and refactoring on Source code.

Test-driven development was introduced as part of a larger software design paradigm known as Extreme Programming (XP), which is part of the Agile software development methodology

It is an iterative development process. In every iteration it starts with set of test cases written for a new piece of functionality.

Test-Driven Development (TDD) Process

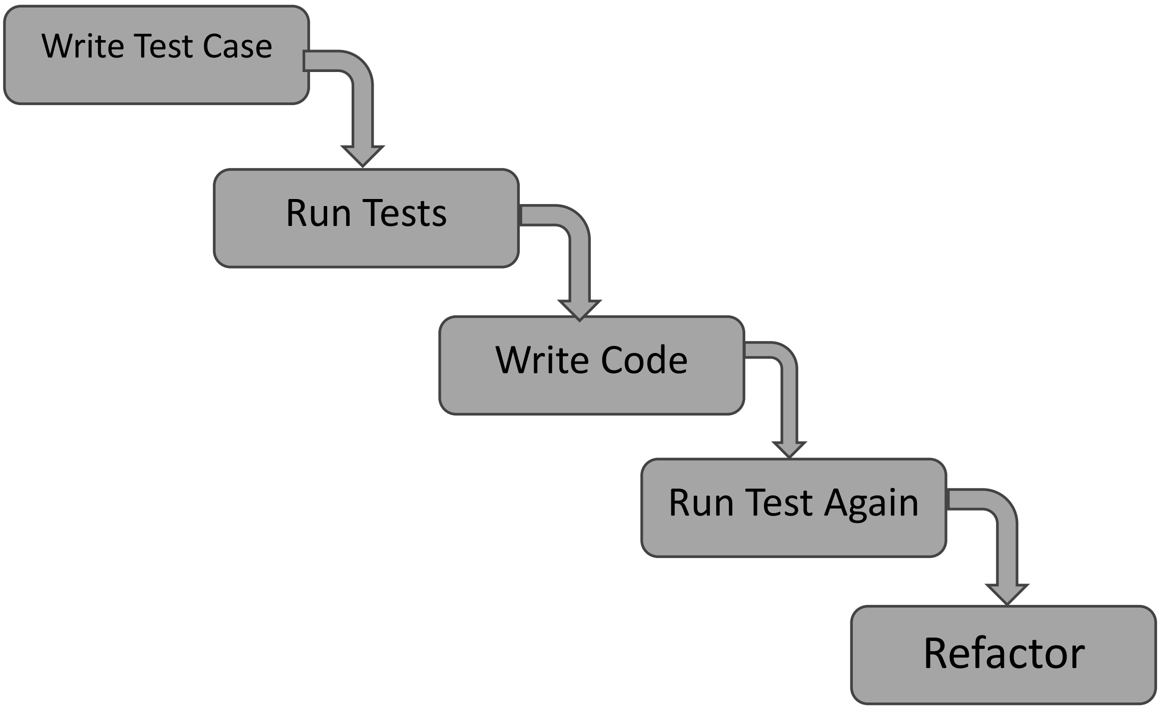


1. **RED:** In this phase, you start by writing a test that defines the desired behaviour or functionality of a specific piece of code. Initially, this test will fail because the corresponding code hasn’t been written yet. This failing test is often referred to as a “RED” test.

2. **Green**: Once you have a failing test, your next step is to write the minimum amount of code necessary to make the test pass. This code may not be perfect or efficient; the goal is to satisfy the test’s conditions and make it pass. When the test passes, it becomes a “green” test, indicating that the desired functionality has been implemented.

3. **Refactor**: After making the test pass, you can improve the code's design, structure, and efficiency while keeping the test green. Refactoring involves making changes to the code without changing its external behaviour. The tests act as a safety net, helping you catch unintended side effects of your changes.

A flowchart showing the steps of TDD



Write Test Cases:

Write tests based on requirements before writing code.

Run Tests:

Run the tests; they should fail initially as there's no code yet.

Write Code:

Write the minimum amount of code necessary to make the tests pass.

Run Tests Again:

After writing code, rerun tests to ensure they pass.

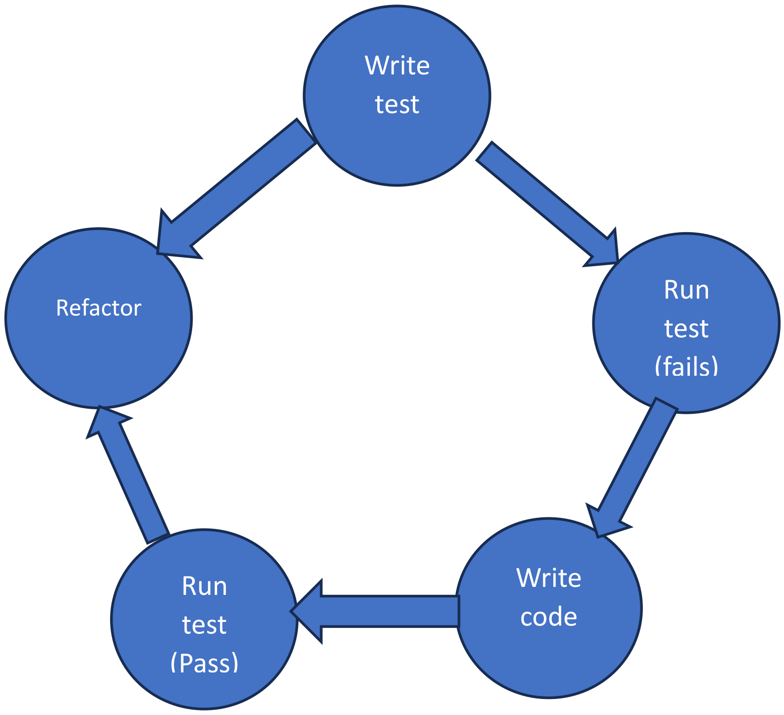
Refactor:

Improve the code without changing its functionality.

Benefits of TDD:

1. **Bug Reduction:**
   - ➢ By catching bugs in the development process, TDD reduces the like hood of bugs in the final product.
2. **Improved Reliability:**
   - ➢ With comprehensive test coverage, software reliability is enhanced, ensuring fewer unexpected issues in production.

TDD Cycle:



Conclusion:

- Test-Driven Development (TDD) is a proven methodology for building robust and reliable software by emphasizing writing tests before code.
- TDD not only reduces bugs but also fosters software reliability through a systematic approach to development.