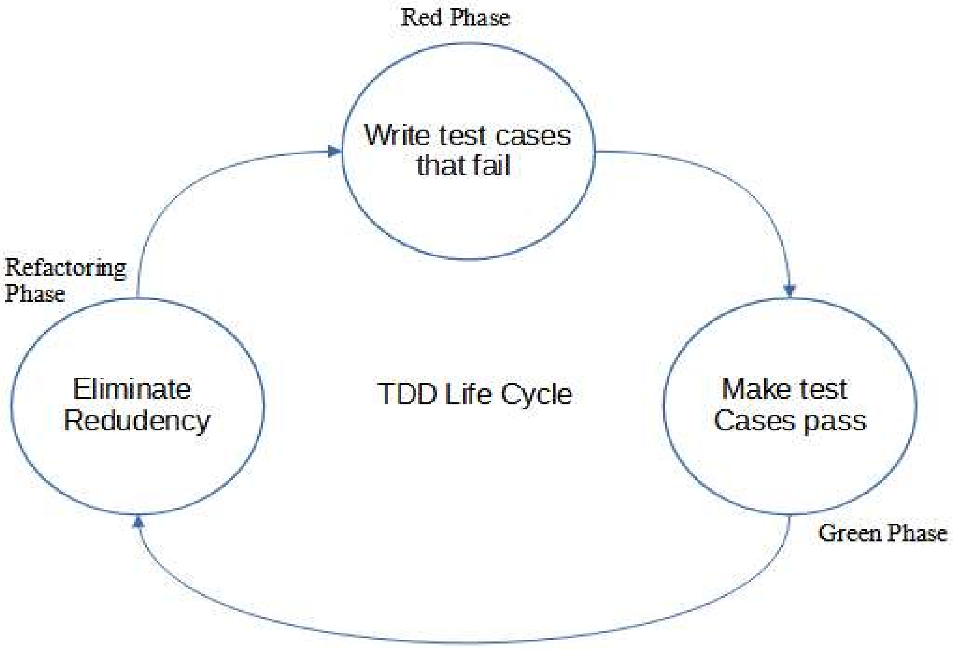
Assignment 1(13-05)

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.

Introduction: Test-driven development (TDD) is an iterative methodology that prioritizes the creation of checking against test cases at every stage of software development, by converting each component of the application into a test case before it is built and then testing and tracking the component repeatedly.



Write a Test: Developers write a test that defines a new function or improvement of an existing function. Run the test to ensure it fails. This confirms that the test is working correctly and that it accurately reflects the current state of the codebase.

Write Code: The developer writes the minimum amount of code necessary to pass the test.Once the code is written, all tests, including the newly added one, are run. If the new test fails, it means the code implementation doesn't meet the desired functionality. The developer then modifies the code to make the test pass.

Refactor Code: After passing the test, the developers refactor the code to improve its structure, readability, or performance while ensuring that all tests continue to pass. The process is repeated for each new piece of functionality.

Benefits of Test Driven Development:

- **Reduced Bugs**: By writing tests before writing code, TDD helps catch bugs early in the development process. Since tests are run frequently, developers can quickly identify and fix issues before they propagate throughout the codebase. This results in more robust and reliable software.

- **Faster Feedback Loop**: TDD provides a rapid feedback loop. When a test fails, developers immediately know that something is wrong and can fix it before moving on to the next task. This reduces the time spent debugging and increases overall productivity.

- **Improved Code Quality**: TDD encourages developers to write modular, well-structured,loosely coupled and easily maintainable code.

- **Regression Testing**: TDD promotes the creation of a comprehensive suite of automated tests. Whenever changes are made to the codebase, developers can quickly run these tests to ensure that existing functionality remains intact. This helps prevent the reintroduction of bugs that may have been previously fixed.

- **Better Documentation**: TDD enhances the productivity of the developer and leads to the development of a codebase that is flexible and easy to maintain.

- **Refactoring**: TDD supports refactoring—the process of restructuring code without changing its external behavior. This encourages continuous improvement and maintenance of code quality without sacrificing reliability.

Conclusion: Test Driven Development is promoting early bug detection, ensuring code quality, facilitating regression testing, clarifying requirements, supporting incremental development, and serving as documentation, TDD significantly contributes to fostering software reliability.