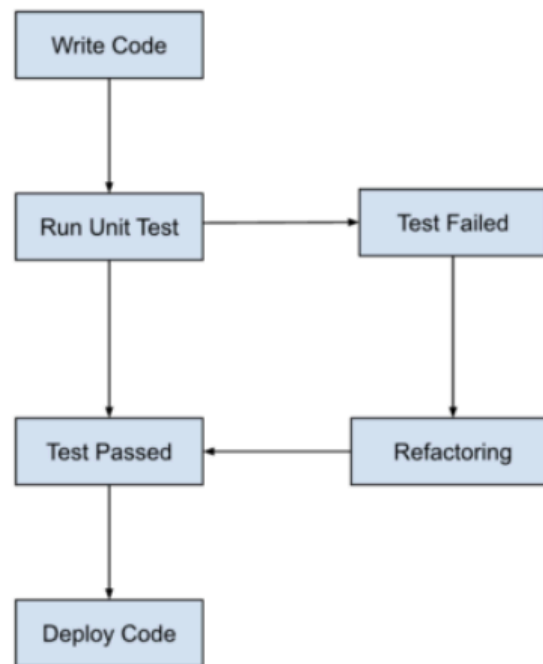


Assignment 1: **Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.**

**Test-driven development (TDD)** is a software development process that follows a specific sequence of steps to ensure quality and reliability in the code. Here's an illustration of the **TDD** process and its benefits:



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

1. **Write a Test:** Before writing any code, the developer writes an automated test case that defines a desired improvement or new function. This test initially fails because the corresponding code has not been implemented yet.
2. **Run the Test:** Execute all tests, including the new one, to ensure that the new test fails. This step validates that the test is correctly set up and that it actually tests what it is intended to test.
3. **Write the Code:** Write the minimum amount of code needed to pass the new test. The code is typically implemented in the simplest way possible to pass the test.
4. **Run All Tests:** Run all tests, including the new one, to ensure that the new code passes all tests and does not break existing functionality.

5. **Refactor (if needed)**: Refactor the code to improve its design without changing its functionality. The tests will ensure that the refactoring does not introduce bugs.

## Benefits of Test-Driven Development (TDD):

- **Early Detection of Bugs**: TDD helps catch bugs early in the development cycle because tests are written before the code, ensuring that all code changes are covered by tests.
- **Improved Code Quality**: The need to write testable code often results in better software design and cleaner code. This reduces the chances of introducing defects and makes the code easier to maintain.
- **Faster Feedback Loop**: TDD provides a fast feedback loop since tests are automated and run frequently. This allows developers to identify and fix issues quickly.
- **Increased Confidence in Code**: By having a comprehensive suite of tests that cover all code paths, developers can be more confident that their changes have not introduced unintended side effects.
- **Better Documentation**: Tests serve as a form of living documentation, demonstrating how the code is intended to be used and what it is supposed to do.
- **Encourages Iterative Development**: TDD encourages an iterative development process, where code is continuously improved and refactored while maintaining functionality.

