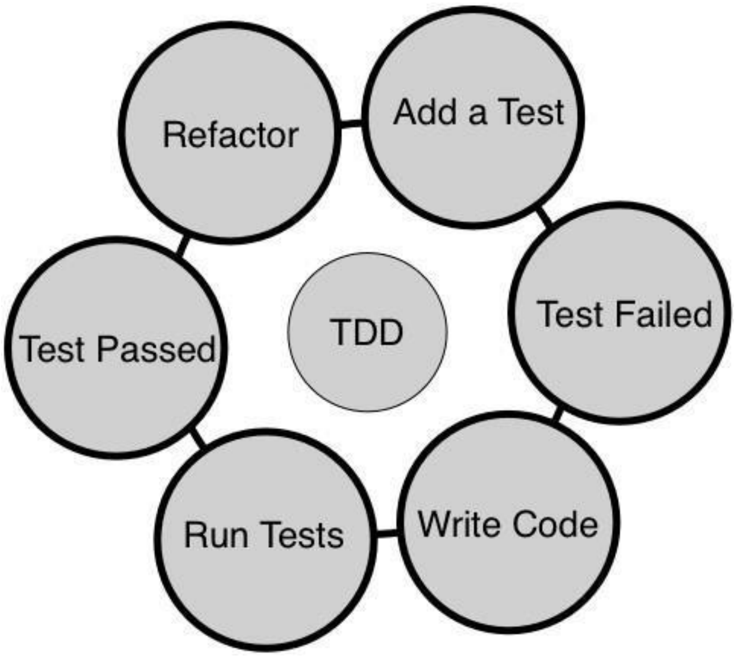
ASSIGNMENT-1

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) Process



1. **Add a Test**:
   - Begin by writing a test case that describes the specific functionality you want to implement.
   - This test case should initially fail because there's no corresponding code yet.

   Benefit:

   By writing tests first, you establish clear expectations for your code, ensuring that it meets the desired requirements.

2. **Test Failed**:
   - After adding the test, run all the existing test cases (including the new one).
   - Since there's no code to make the new test pass, it will fail at this stage.

Benefit:

Identifying failures early allows you to catch potential issues before they propagate further into your codebase.

3. **Write Code**:
- Develop the minimum amount of code necessary to make the test case pass.
- Keep the code simple and focused on the specific functionality being tested.

Benefit:

- TDD encourages incremental development, leading to more modular and maintainable code.

4. **Run Tests**:
- Execute all the test cases again, including the new one.
- If any test fails, revisit the code and make necessary adjustments.

Benefit:

Regularly running tests ensures that your code remains functional and aligned with requirements.

5. **Test Passed**:
- Once all the tests pass, you’ve successfully implemented the desired functionality.
- Congratulations! Your code now meets the specified requirements.

Benefit:

TDD provides confidence that your code behaves correctly, reducing the risk of introducing defects.

6. **Refactor (Optional)**:
- After passing the tests, consider refactoring your code.
- Refactoring involves improving the code’s structure, readability, and efficiency without changing its behavior.

Benefit:

Refactoring maintains code quality, making it easier to maintain and extend.