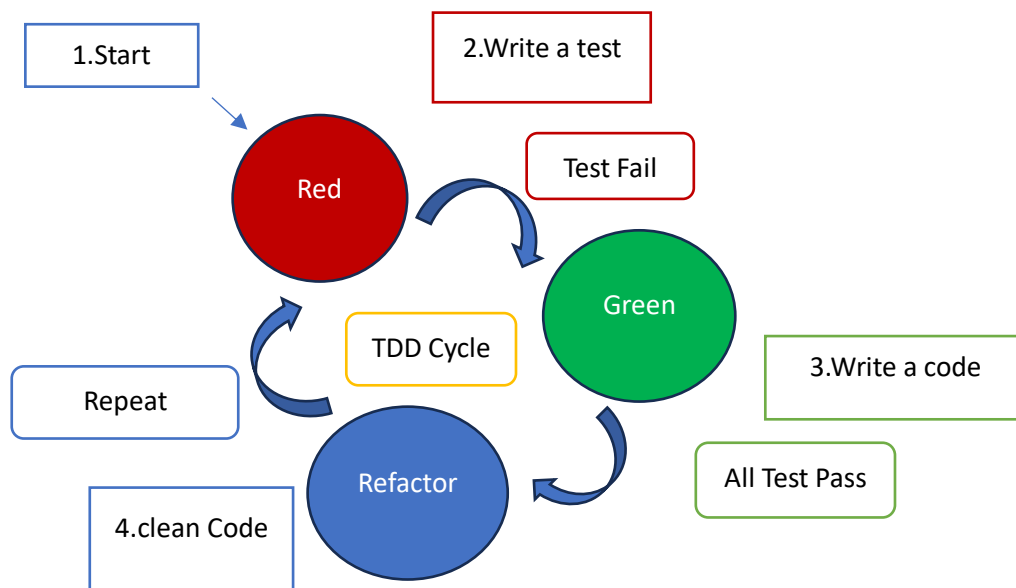# DAY 3 ASSIGNMENTS:

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?

I have Done the Diagram in word with the Help of Shapes.

Infographic: Test-Driven Development (TDD):

- Test-Driven Development (TDD) is a software development approach where tests are written before the actual code.
- It is a Step-by-Step guide.

The TDD Cycle:

1. **Start:**
   It sends the feature Request.

2. **Write a test:**
   Start by writing a test for the next bit of functionality.

3. **Test fail:**
   Running the test to see if it fails and it should fail the functionality is not implemented yet.

4. **Write Code:**
   Write the code to pass the test functionality.

5. **Run all tests:**
   Run all the test cases to ensure that the new code does not break the current functionality.

6. **Refactor:**
   Refactor is to improve the structure of the code without changing its external behaviour.

7. **Repeat:**
   This will repeat with further improvement and new features until the product is complete.

Benefits of TDD (Test-Driven Development):

1. **Bug Reduction:**
   - It promotes a fail-first approach where issues are identified immediately after the test fails.
   - Preventing Bugs from reaching production

2. **Improve code quality**:
   - The Code is tested fully, cleanly written and maintainable.

3. **Software Reliability**:
   - TDD helps in catching bugs early in the development cycle.

4. **Fast Development**:
   - It Reduces Debugging Time.
   - Issues Can be fix Later.
   - It Catches Bugs Early.

5. **Test Coverage:**
   - It has High Test coverage that ensuring most part of code is tested.
   - This leads to become a full software reliability.

6. **Clear Requirement and Design**:
   - Writing a test before code encourage developers to define clear requirement and design.
   - It is More flexible.

Advantages of Test-Driven Development (TDD):

- Easy to maintain.
- Easy to refactor.
- Better design.
- High test coverage.
- Less debugging.
- Improve productivity.

Disadvantage of Test-Driven Development (TDD):

- Slow process.
- Extra Maintenance.
- Not Always Applicable.