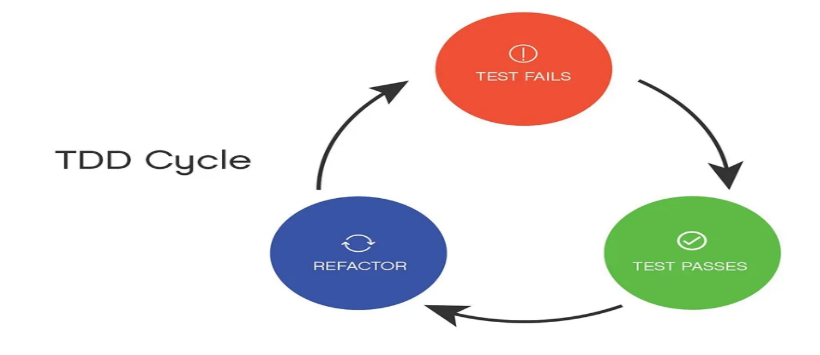
Day 3 – Assignment 1

* 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. **What is TDD?**

* A software development process where you write tests before writing the actual code.
* Focuses on writing small, incremental tests that define desired improvements or new functions.

1. **TDD Cycle**



1. **Write a Test**

* Identify a new feature or function.
* Write a test that defines the desired behavior.
* The test should initially fail since the functionality isn't implemented yet.

1. **Run All Tests**

* Execute all tests in the suite.
* Confirm that the new test fails, verifying that the test is valid.

1. **Write Code**

* Write the minimal amount of code necessary to make the test pass.
* Focus on simplicity and clarity.

1. **Run Tests Again**

* Execute all tests to see if the new code passes the test.
* Ensure no existing functionality is broken.

1. **Refactor Code**

* Improve the code structure without changing its behavior.
* Enhance readability, maintainability, and performance.

1. **Repeat**

* Continue the cycle for the next feature or improvement.

1. **Benefits of TDD**

* **Bug Reduction**
* Early detection and fixing of defects.
* Ensures code works as expected from the beginning.
* **Software Reliability**
* Continuous testing assures the software’s robustness.
* Enhances confidence in the codebase.
* **Improved Design**
* Promotes simple and modular code design.
* Facilitates better architecture and easier modifications.
* **Documentation**
* Tests act as living documentation for the code.
* Makes it easier to understand code behavior.

1. **Best Practices**

* **Keep Tests Small and Focused**
* Each test should check a single behavior or function.
* **Use Descriptive Test Names**
* Names should clearly describe what the test is validating.
* **Maintain Test Independence**
* Tests should not depend on each other.
* Ensures reliability and ease of debugging.
* **Run Tests Frequently**
* Integrate test execution into the development process.
* Helps catch issues early.

1. **Conclusion**

* **Adopting TDD**
* Embracing TDD can lead to better software quality and more efficient development cycles.
* Encourages a proactive approach to development and maintenance.