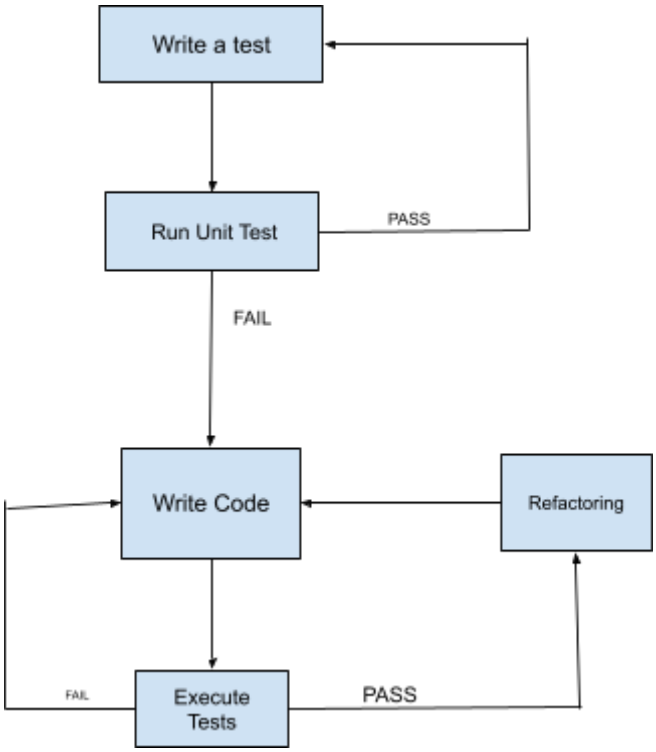
Day3 Assignment
Assignment 1:

Infographic representation of Test Driven Development:



The steps involved in TDD are listed below:
1. Write Tests First: Begin by creating tests for the desired functionality before writing any code.
2. Run Tests: Execute the tests to ensure they fail initially. This confirms that the tests are correctly assessing the absence of the desired functionality.
3. Write code: Develop the minimum amount of code required to pass the failing tests.
4. Execute the tests again: Re-run the tests to validate the new code which should pass.
4. Refactor code: Improve the structure and design of the code without altering its functionality. Ensure tests continue to pass after refactoring.

Benefits of TDD:

1. Bug Reduction: By writing tests first, developers catch bugs early in the development process, reducing the likelihood of bugs in the final product.
2.
3. Improved Software Reliability: TDD encourages developers to write reliable and maintainable code since each piece of functionality is thoroughly tested before implementation.

4. Enhanced Design: TDD promotes better code design as developers focus on writing code that is modular, loosely coupled, and easy to maintain.

5. Faster Development: While it may seem counterintuitive at first, TDD can lead to faster development cycles because bugs are caught early, reducing the time spent debugging later in the process.

Assignment2:

PARAMETERS	TDD	BDD	FDD
Approach	Write tests before writing code. Develop code incrementally to pass tests.	Focus on behavior of the system from an end-user perspective. Write tests in natural language.	Break down development into features,each with its own development cycle.
Benefits	1. Early bug detection. 2. Improved code reliability. 3. Enhanced Design through Iterative development.	1. Encourages collaboration Between stakeholders. 2. Promotes a shared understanding of requirements.	1. Scalability of large projects. 2. Clear focus on features and deliverables. 3. Promotes incremental development.
Suitability	Ideal for projects requiring frequent code changes where reliability is critical.	Ideal for projects with complex Business requirements and diverse stakeholders.	Ideal for large scale projects with clear feature- based deliverables.