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.

=============================================================

TDD:- Test-driven development (TDD) is a method of coding in which you first write a test and it fails, then write the code to pass the test of development, and clean up the code. This process recycle for one new feature or change. In other methods in which you write either all the code or all the tests first, TDD will combine and write tests and code together into one.

1. “Add a check” instead of “Add a test”
2. “Run all checks” instead of “Run all tests”
3. “Do the work” instead of “Write some code”
4. “Run all checks” instead of “Run tests”
5. “Clean up the work” instead of “Refactor code”

Process of Test Driven Development (TDD) :-

It is the process in which test cases are written before the code that validates those cases. It depends on the repetition of a concise development cycle. Test-driven Development is a technique in which automated Unit tests are used to drive the design and free decoupling of dependencies. The following sequence of steps is generally followed:

Step 1. Write a complete test case describing the function. To make the test cases the developer must understand the features and requirements using user stories and use cases.

Step 2. Run all the test cases and make sure that the new test case fails.

Step 3. Write the code that passes the test case

Step 4. Run the test cases

Step 5.Refactor code – This is done to remove duplication of code.

Repeat the steps again and again.

### 

### Three Phases of Test Driven Development ( TDD ):-

* 1. Create precise tests: Developers need to create exact unit tests to verify the functionality of specific features. They must ensure that the test compiles so that it can execute. In most cases, the test is bound to fail. This is a meaningful failure as developers create compact tests based on their assumptions of how the feature will behave.
* 2. Correcting the Code: Once a test fails, developers must make the minimal changes required to update the code to run successfully when re-executed.
* 3. Refactor the Code: Once the test runs successfully, check for redundancy or any possible code optimizations to enhance overall performance. Ensure that refactoring does not affect the external behavior of the program.

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

1. Fosters the creation of optimized code.

2. It helps developers better analyze and understand client requirements and request clarity when not adequately defined.

3. Adding and testing new functionalities become much easier in the latter stages of development.

4. Test coverage under TDD is much higher compared to conventional development models. The TDD focuses on creating tests for each functionality right from the beginning.

5. It enhances the productivity of the developer and leads to the development of a codebase that is flexible and easy to maintain.