Assignment 1

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.

Infographic: Test-Driven Development (TDD) Process

1. **Introduction to TDD:**

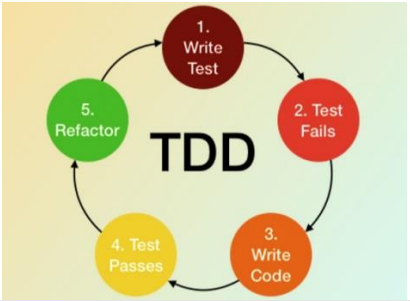
- Brief definition of TDD.

TDD, or Test-Driven Development, is the process of writing and executing automated tests before writing code. Insights from the tests help the developer improve the code. TDD reflects the spirit of continuous feedback, resulting in faster bug identification and debugging.

- Importance of writing tests before code.

2. **TDD Process Overview:**

- Visual representation of the TDD cycle:
    - Write Test
    - Run Test (Fail)
    - Write Code
    - Run Test (Pass)
    - Refactor



3. **Step 1: Write Test:**

- Image/icon of a developer writing a test code.
- Write a test case that describe the function completely. In order to make the test cases the developer must understand the features and requirements using user stories and use cases.

4. **Step 2: Run Test (Fail):**

- Image/icon of a test failing.
- Run all the test cases and make sure that the new test case fails.

5. **Step 3: Write Code:**

- Image/icon of a developer writing the actual code to make the test pass.
- Write the simplest code to make the test pass. Avoid over-engineering.

6. **Step 4: Run Test (Pass):**

7.

- Image/icon of a test passing.

Run the test again. It should pass now, indicating that the code fulfills the requirements specified by the test.

8. **Step 5: Refactor:**

- Image/icon of a developer refining the code without changing its behavior.
- Refactor the code to improve its design, readability, and performance without altering its functionality.

9. Repeat the steps again and again

TDD is a continuous, iterative process of improving code through tests. To do TDD, simply follow the mantra of **Red - Green - Refactor cycle**. Some may call it **Fail - Pass - Refactor**, but it's all the same thing.

1.Red – Create a test case and make it fail
2.Green – Make the test case pass by any means.
3.Refactor – Change the code to remove duplicate/redundancy.

10. **Benefits of TDD:**

- Image/icon of a checklist with checkmarks.
- Highlights of TDD benefits:
    - Early bug detection.
    - Improved code quality.
    - Faster development cycles.
    - Increased software reliability.
    - Easier code maintenance and refactoring.
    - Unit test provides constant feedback about the function
    - Quality of design increases which further helps in proper maintenance.
    - Test driven development act as a safety net against the bugs
    - TDD ensures that your application actually meets requirements defined for it.
    - TDD have very short development lifecycle.

11. **Conclusion:**

- Summarize the key points of TDD.
- Encourage adoption of TDD for building robust and reliable software.