Name: V. Charan Kumar

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:

Test-Driven Development (TDD) is a methodology in software development that focuses on an iterative development cycle where the test cases are written before the actual feature or function is written.

Step 1: Write Tests

Write testcases before writing any code. Tests define the expected behavior of the code.

Step 2: Run Tests

Run the tests to verify. Failing tests indicate that the code needs to be implemented or fixed.

Step 3: Write Code

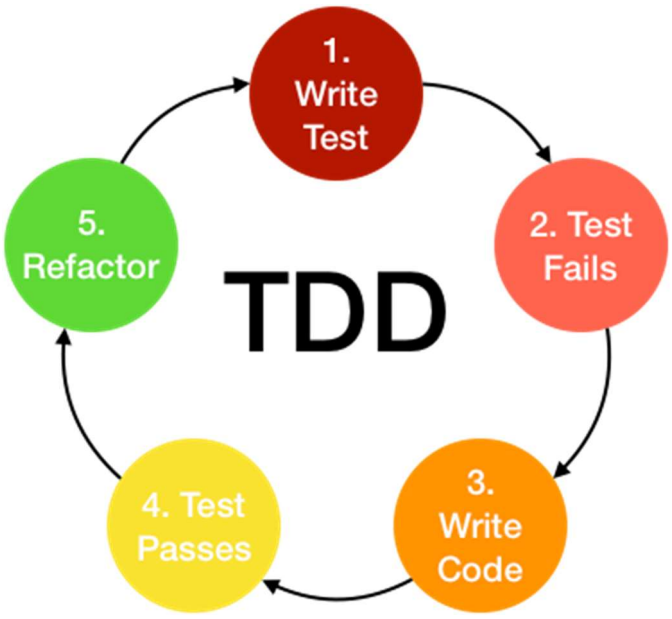
Implement the code to make the failing tests pass. Write only the code necessary to fulfill the requirements of the testcases.

Step 4: Run Tests Again

Run the tests again to ensure that the implemented code passes. Passing tests indicate that the code behaves as expected and meets the requirements.

Benefits of TDD:

1. **Bug Reduction:**
    - By writing tests first, developers catch bugs early in the development process, reducing the likelihood of introducing defects.
2. **Improved Code Quality:**
    - TDD encourages writing modular, maintainable code that is easier to understand and refactor.
3. **Faster Feedback Loop:**
    - Automated tests provide immediate feedback on the correctness of the code, allowing developers to make quick iterations.
4. **Increased Confidence:**
    - TDD fosters confidence in the codebase by ensuring that changes do not introduce regressions or unintended side effects.
5. **Software Reliability:**
    - By continuously testing and refining the code, TDD leads to more reliable and stable software products.



Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

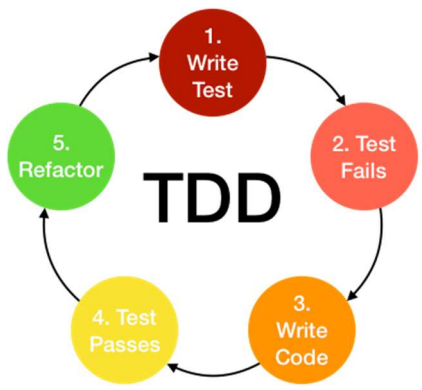
Test Driven Development:

Definition: Here tests are written before code implementation.

Approach: Write test cases first, then implementing code to pass those tests.

Benefits: Improved code quality, bug reduction, and faster feedback loops.

Suitability: suitable for projects that require more testing and frequent code changes.



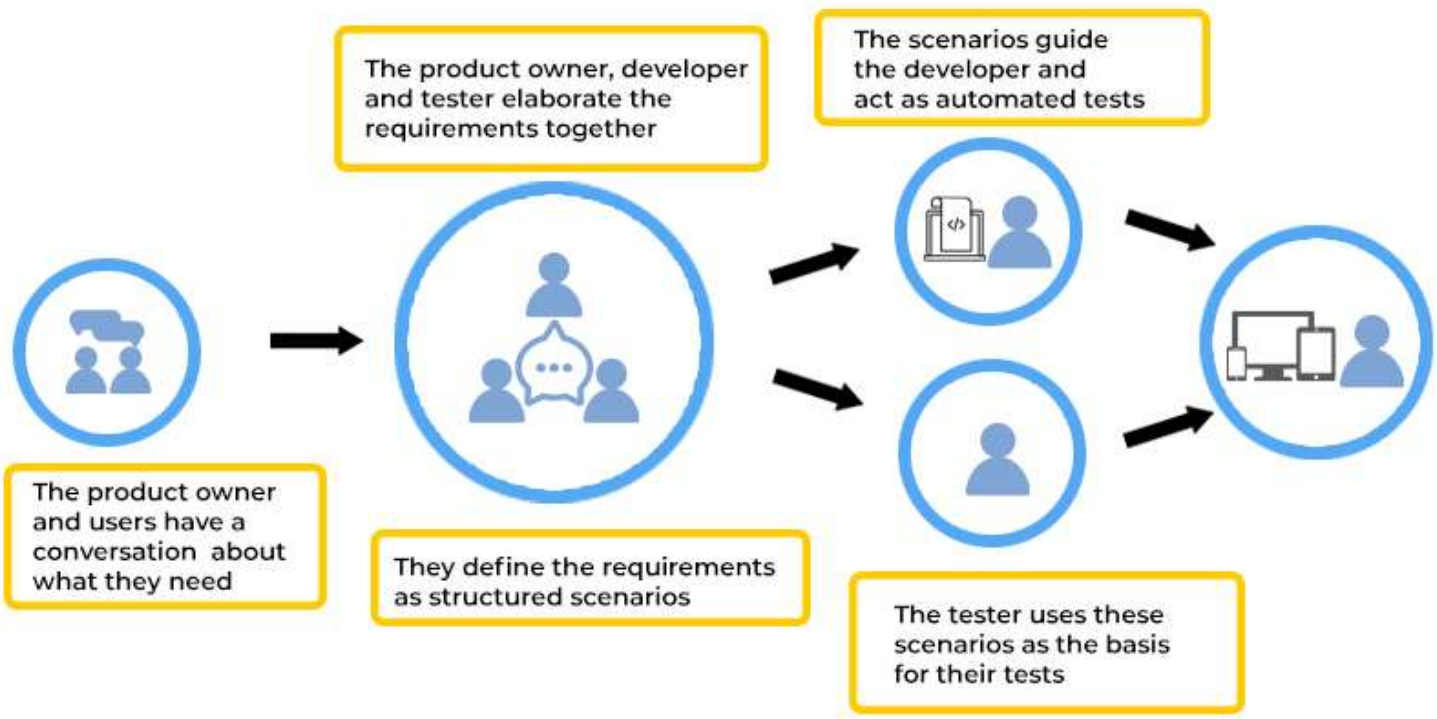
Behaviour Driven Development:

Definition: Understand the behaviour of the system from the end-user perspective.

Approach: breaking down the project into features, designing them, and building them iteratively.

Benefits: improved collaboration between stakeholders, clearer requirements, and alignment with business goals.

Suitability: BDD is suitable for projects with complex business requirements and need for stakeholder involvement.



Feature Driven Development:

Definition: Iterative and incremental approach focused on building features incrementally.

Approach: defining system behaviour using human-readable scenarios or examples.

Benefits: clear progress tracking, reduced complexity.

Suitability: FDD is suitable for projects requiring feature-based development and early feature delivery.