# Test-Driven Development (TDD)

1. **Write Test Cases First**
   * **Step 1:** Write failing test cases based on requirements.
   * **Step 2:** Specify expected outcomes for each test.
2. **Implement Code**
   * **Step 3:** Write code to pass the failing tests.
   * **Step 4:** Focus only on writing enough code to pass tests.
3. **Refactor Code**
   * **Step 5:** Refactor code for clarity and performance.
   * **Step 6:** Ensure all tests pass after refactoring.

**Benefits of TDD**

* **Bug Reduction:** Detects bugs early in the development process.
* **Improved Design:** Encourages modular, cleaner code.
* **Higher Confidence:** Each passing test enhances reliability.
* **Documentation:** Tests serve as live documentation.

**Impact on Software Reliability**

* **Fewer Bugs:** Early bug detection prevents issues later.
* **Maintainability:** Easier to update and modify code.
* **Quality Assurance:** Continuous testing ensures robustness.

**Conclusion**

Test-Driven Development (TDD) ensures software reliability through rigorous testing and iterative development. By prioritizing tests from the outset, developers create more resilient and maintainable code.

