**Test-Driven Development (TDD) Process**

**Key Steps in TDD**

**Define Requirements** → Break features into small, testable units.  
**Write Test First** → Ensures code meets specs from the start.  
**Run Test (Fail Expected)** → Verify the test fails (confirms validity).  
**Implement Code** → Write *just enough* to pass.  
**Run All Tests** → Ensure no regressions.  
**Refactor & Optimize** → Clean up while keeping tests green.

**Benefits of TDD**

**Early Bug Detection** → Fail fast, fix fast.  
 **Cleaner Code** → Forces modular, maintainable design.  
 **Living Documentation** → Tests describe system behavior.  
 **Confidence in Changes** → Refactor fearlessly with test safety nets.  
 **Faster Debugging** → Isolate issues via small test cases.

**Pro Tips**

**Keep tests small & focused** → Test one behavior per case.  
 **Use descriptive names** → e.g., testCalculateDiscount\_ExceedsBudget().  
 **Mock dependencies** → Isolate units for reliable testing.

**Bottom Line**

TDD shifts focus to **prevention over cure**, ensuring robust, scalable software.