Pragmatic Unit Testing – CheatSheet 1.1

Right-BICEP		
Right	Are the results **right?**	Use data files, golden bits, oracle…
B	Are all **boundary** conditions CORRECT	**C**onformance, **O**rdering, **R**ange, **R**eference, **E**xistence, **C**ardinality, **T**ime.
I	Can you check **inverse** relationships?	Apply logical inverse (e.g. sqrt vs pow).
C	Can you **cross-check** results using other means?	Use alternative way of achieving result.
E	Can you enforce **error conditions** to happen?	What errors could occur, e.g. environmental constraints.
P	Are **performance** characteristics within bounds?	Quick regression test of performance characteristics.

CORRECT		
C	Conformance	Any specific data format? What if the format is different?
O	Ordering	Order of data or position of an element? Reverse order?
R	Range	Start and end of indices, first is greater than last, index is negative, index is greater than allowed, count doesn't match number of actual number of items…
R	Reference	Any external dependencies? Any preconditions? Do we guarantee post-conditions?
E	Existence	Does some given thing exist? Null, blank, empty, 0 (zero)?
C	Cardinality	12 feet lawn, each section 3 feet, how many poles required? 4? Actually 5! Test for how many things there might be: Zero, one or more than one: 0-1-n-Rule.
T	Time	Relative time (ordering in time), absolute time (elapsed time), concurrency issues?

A-TRIP (Good Tests)		
A	Automatic	Running the test and checking the results should be automatic.
T	Thorough	Well-tested methods may have 4-5 asserts.
R	Repeatable	Run over and over again, in any order produce the same results.
I	Independent	Keep tests tight, focused, test only one thing at once.
P	Professional	Write real code, refactor.