**CHECKLIST: Test Cases**

❑ Does each requirement that applies to the class or routine have its own testcase?

❑ Does each element from the design that applies to the class or routine have its own test case?

❑ Has each line of code been tested with at least one test case? Has this been verified by computing the minimum number of tests necessary to exercise each line of code?

❑ Have all defined-used data-flow paths been tested with at least one test case?

❑ Has the code been checked for data-flow patterns that are unlikely to be correct, such as defined-defined, defined-exited, and defined-killed?

❑ Has a list of common errors been used to write test cases to detect errors that have occurred frequently in the past?

❑ Have all simple boundaries been tested: maximum, minimum, and off-byone boundaries?

❑ Have compound boundaries been tested—that is, combinations of input data that might result in a computed variable that’s too small or too large?

❑ Do test cases check for the wrong kind of data—for example, a negative number of employees in a payroll program?

❑ Are representative, middle-of-the-road values tested?

❑ Is the minimum normal configuration tested?

❑ Is the maximum normal configuration tested?

❑ Is compatibility with old data tested? And are old hardware, old versions of the operating system, and interfaces with old versions of other software tested?

❑ Do the test cases make hand-checks easy?