Code Check List for Effective Personal Reviews and Formal Inspections
W.L.Honig
Version 1.0
27Sept 08

A basic check list for code. Your team can extend if appropriate and necessary for your quality plan goals. Keep your checklists under configuration control and be sure all use the "latest" one.

Style Guide Defects (Style Defects may be minor or major)
1. Check everything in the style guide!

(many of these not only improve readability but also find or lessen chance of bugs if used properly!)

Possible Defects
1. Are size and bounds checks carried out when and where necessary? (a good practice is to use inclusive lower limits and exclusive upper limits)
    - Yes,
2. Is there ever any mismatch between the units of an expression and those of a variable? (feet to meters)
    - No, in cases where types could possibly be mismatched they were cast or otherwise converted into the correct type.
3. Are number variables appropriate maximum size? (int, double, etc.)
    - Yes, number variables were chosen well, typically using the simplest type possible to represent the data.
4. Are array, buffer, table (etc.) sizes appropriate? (Too little or too big, both bad)
   Yes, all data structures used are of appropriate size for the amount of data they contain.
5. Are equality tests used on floating point numbers? (Always bad)
   No, there are never any equality tests using floating point numbers.
6. When testing for errors, could the "error condition" actually be legitimate in some cases?
   Yes, there is an issue with the parent ID variable from the parent class that causes an error condition to be legitimate.
7. Are exceptions declared, thrown and handled correctly?
   Yes, almost all necessary exceptions have been created, except in some cases where a value is passed to an if statement that cannot be evaluated.
8. Is every function call the correct function? (Not a similarly named function)
   Yes, most if not all functions have different names, and none of them are used in the wrong place.
9. Are preconditions checked before a function call (if necessary)?
   Preconditions are checked, but need to be added to notes.
10. Are postconditions ensured at the time of every possible function return point?
    All postconditions are ensured and each function returns the proper type.
11. Are files and other resources closed in all possible exits?
    All files and resources are closed when possible.

Note:
Studies have shown that the most acceptable forms of for loops (to avoid off-by-one errors) are these:

```
for ( i = 0; i <= max_index; ++i )
for ( i = 0; i < sizeof(array); ++i )
for ( i = max_index; i >= 0; --i )
for ( i = max_index; i; --i )
```