**Code Review Checklist**

Design:

Class

1. Scope of the class
2. Name of the class
3. Number of lines/functions
4. Constructor – Default/Overloaded/Testable
5. Destructor
6. Thread safe
7. Proper indentation
8. Proper comments
9. Proper namespace
10. Unwanted namespace imports
11. Avoid nested classes
12. Unused classes
13. Deprecated classes

Members

1. Member variables
2. Access specifier
3. Datatype
4. Name
5. Thread safe
6. Read only/Const
7. Initialization of members

Methods

1. No. of lines
2. Name
3. Comments
4. Error/Exception handling
5. Try/catch block should have ‘finally’ for proper cleanup
6. Logging
7. Localization
8. Deprecated methods
9. Unused methods
10. Proper Return type
    1. Parameters
       1. No. of parameters
       2. Order of parameters – In, Out, Optional
       3. Parameter validation
       4. Unused parameters
       5. Datatypes (Avoid Type Casting)
       6. Proper default parameters
    2. Conditionals
       1. Logic inside conditionals – Can it be moved to diff class/function
       2. No nested conditionals
       3. Usage of proper conditional construct (If/Switch/Ternary operator)
       4. Order of conditional execution (AND/OR logics)
       5. == checking
       6. Make sure object is initialized before use
    3. Assignment
       1. No magic numbers
       2. Use Macros/Enums/const variables
       3. Make sure object is initialized before use
       4. Take care of type casting/buffer overrun..
    4. Iteration
       1. No infinite loops
       2. Iteration variable to be initialized
       3. Iteration variable name should be proper
11. Events/Delegates
12. Unit test
    1. Mocking
    2. Dependency injection
    3. 100% code coverage for public methods
    4. Happy flow
    5. Negative flow
       1. Exceptions
       2. All possible return values
       3. Parameter validations – Upper bound/ Lower bound/ Default/ Null
       4. All possible values for out parameters
13. Is member variable relevant?
    1. Based on responsibility and extensibility of the class, Required across multiple functions;
    2. Memory consequences
14. Access specifiers