Code review Findings:

Extract constants into one class

2. Favour smaller interfaces over larger ones

3. Ensure code coverage % is met

4. Remove unused members

5. Always close streams

6. Keep classes small and concise

7. Classes should be structured in the following way (Instance variable, constructors, methods)

8. Remove old comments

9. Code should be comment light and self-documenting

10. Approach returning null from a method call with extreme caution

11. Don’t repeat yourself (Refactor code instead of copy paste)

13. If we have more than a few nested logic statements refactor out appropriate methods

14. Constructors should never normally have more than five parameters

15. Ensure encapsulation is properly followed (instance variables private with public getters and setters)

16. Members should be properly named (A variable called a or b is not self-documenting )

17. Proper catch blocks should be added for exception handling instead of single Exception object handler.

18. Uniform coding standards for braces, loops, if-else, switch should be used across the application. This will usually be company specific.

19. Similar logic should be declared in a helper class and called from multiple places.

20. Usage of API classes and methods should be encouraged instead of writing custom code for performing the same operations. Don’t reinvent the wheel

21. A single statement should not go beyond the viewable area of the editor or IDE and should be split across multiple lines.

22. The addition of any piece of code should not break existing functionality.

23. Creating immutable class should be encouraged when appropriate.

24. The best way to check if the String object is neither null nor empty string is to use the following code: if(“”.equals(str))