INB370 Assignment 1 Code Quality Marking Guide

This guide provides a checklist-style statement of the code quality criteria that will be used to assess your code. We recommend *strongly* that you take advantage of the opportunity provided by the pracs to get some feedback from your tutors prior to submission.

Assessment Criteria

- ☐ **Layout.** All program code and unit tests are neatly laid out with appropriate use of whitespace and avoidance of overly wide lines.
- ☐ **Presentation.** All comments and other free-form text, such as exception messages, are clear, well-written and free of spelling or grammatical errors.
- ☐ **Readability.** Meaningful identifiers are used for all methods, parameters, fields, etc, *including loop indices*. The meaning of each identifier is made clear by avoiding cryptic abbreviations and other such obfuscation.
- ☐ **Simplicity.** Data structures, algorithms and control flow constructs are all as simple as possible.
- ☐ **Maintainability.** The program code and unit tests are all made easy to maintain by use of named constants instead of 'magic numbers'. Methods are small, each serving a single purpose. In particular, each unit test serves one purpose only.
- ☐ **Documentation.** The program and unit tests are appropriately commented, in a way that complements the code, by explaining *what* the code does or *why*, but does not merely repeat *how* it works (which should be self-evident from the code itself). Commenting is sparse, serving to clarify unclear code segments only. (NB: There is no need to duplicate the given Javadoc comments.)

In the event we are concerned about plagiarism or otherwise unsure of a student's command of an assignment, we reserve the right to require a student to explain their code – including the set of unit tests.