APPENDIX A: Code Documentation Template

FILE

```
/*
============================================================================
 FILE        : filename.extension
 AUTHOR      : author name
 DESCRIPTION : brief description of content/purpose of the file
 COPYRIGHT   : created
 REVISION HISTORY
 Date:               By:           Description:
 revision date      author         description of the change
 revision date      author         description of the change
 .
 .
 .
 revision date      author         description of the change
============================================================================
*/
```

FUNCTION

```
 /*
============================================================================
 FUNCTION    : function name
 DESCRIPTION : brief description of the what the function does
 ARGUMENTS   : (list all parameters pass in the function)
               dataType variableName - brief description
               dataType variableName - brief description
               .
               .
               .
               dataType variableName - brief description
 RETURNS     : (no return) void
               (with return)dataType - brief description
===========================================================================
*/
```

APPENDIX B: Rubric Scoring

Rubric 1: Assessing and evaluating ability to construct data structures and algorithm consistent with common engineering design situations.

Expert (1.0)	Proficient (1.3)	Competent (2.0)	Novice (3.0)	Beginner (4.0)	(5.0)
▪ The solution is completely accurate. ▪ Demonstrates a thorough understanding of the tradeoffs for algorithms employing different data structures and/or design strategies for given problems.	▪ The solution is overall correct but with few computational errors. ▪ Demonstrates a thorough understanding of the tradeoffs for algorithms employing different data structures and/or design strategies for given problems.	▪ The solution is overall correct but with few computational errors. ▪ Demonstrates a near complete understanding of the tradeoffs for algorithms employing different data structures and/or design strategies for given problems.	▪ The solution is incorrect in several places. ▪ Demonstrates an incomplete understanding of the tradeoffs for algorithms employing different data structures and/or design strategies for given problems.	▪ The solution is done incorrectly. ▪ Demonstrates lack of understanding of the tradeoffs for algorithms employing different data structures and/or design strategies for given problems.	▪ No solution at all.

Rubric 2: Assessing and evaluating ability to implement data structure and algorithm appropriate for solving a given scenario

Expert (1.0)	Proficient (1.3)	Competent (2.0)	Novice (3.0)	Beginner (4.0)	(5.0)
Code executes flawlessly satisfying all of requirements 1 to 5 1. Code submitted on time 2. Used the best algorithm and data structure 3. Code is modular and well structured 4. Exception handling implemented 5. Conforms to code documentation standards	Code executes satisfying requirements 2 to 5	Code executes satisfying partially requirement 2 and ALL of requirement 3 to 5	Code executes satisfying partially requirement 2 and AT LEAST TWO of requirements 3 to 5	Code executes satisfying partially requirement 2 and ONLY ONE of requirements 3 to 5	Code does not execute at all, even if requirements 3 to 5 are satisfied or not