SOEN6011 Deliverable 3: Problem 5 and Problem 7

Anqi Wang 40057695

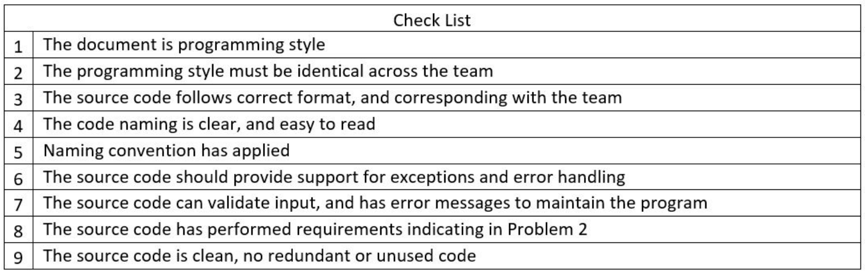
Github Repo address:
https://github.com/AnqiAngelineWang/SOEN6011

1 Introduction

This documentation is working on the source code review of Function 9, and the test case review for Function 10.

2 Question 5: Source code review of Function 9

The source code review is a necessary step for a program to be successful. It is essential to find problems and reduce future risks. Code review may also help with comparing requirement functionally, and improve quality to become robust. This source code view contains manual check and automatic tools check. I have used two sets of automatic tools: Checkstyle Tool and Codacy Tool as guidelines to conduct the source code review. The check list has shown below.



Check List	
1	The document is programming style
2	The programming style must be identical across the team
3	The source code follows correct format, and corresponding with the team
4	The code naming is clear, and easy to read
5	Naming convention has applied
6	The source code should provide support for exceptions and error handling
7	The source code can validate input, and has error messages to maintain the program
8	The source code has performed requirements indicating in Problem 2
9	The source code is clean, no redundant or unused code

Table 1. Source code review guidelines

Comments:

1. For manual check, this code is clean, the logic is clear and in good manner. Also, it follows the team's coding standard and programming style. I have

requested Problem 2 for requirements. After comparing it with the source code, it can be deduced that the source code has included functionalities, and most of requirements have achieved.

2. For automatic tools check, Checkstyle Tool and Codacy Tool indicates some formats that are not regulated:
1). Some imported user interfaces are never used.
2). Some imported user interfaces have wrong lexicographical order.
3). Some indentation formats are not correct. It contains tab characters.

Overall, the source code is readable, maintainable, secure with error handling and messages.

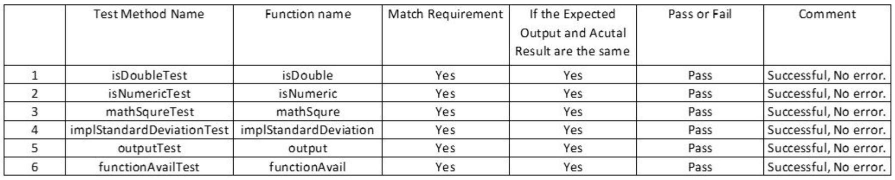
3 Question 7: Test case review for Function 10.

1. Set up computing environment

Test cases are important to check program's accuracy. I have imported the Function 10's project into my Java Coding Eclipse IDE. The test cases applied under Junit 5. In order to proceed with launch for the program test cases successfully, I have configured Junit Testing into my Eclipse IDE, and set up corresponding Java version to meet the computing environment.

2. Requirements and Test cases

Reference to requirements functionalities in Problem 2, each of the test case is clearly associated with the requirement. There are 6 test cases. All test cases are passed. There is no error or failure. The accuracy is 100 %. The test cases have covered and match with all functions related to the source code. The source code coverage is 100 %.



	Test Method Name	Function name	Match Requirement	If the Expected Output and Acutal Result are the same	Pass or Fail	Comment
1	isDoubleTest	isDouble	Yes	Yes	Pass	Successful, No error.
2	isNumericTest	isNumeric	Yes	Yes	Pass	Successful, No error.
3	mathSqureTest	mathSqure	Yes	Yes	Pass	Successful, No error.
4	implStandardDeviationTest	implStandardDeviation	Yes	Yes	Pass	Successful, No error.
5	outputTest	output	Yes	Yes	Pass	Successful, No error.
6	functionAvailTest	functionAvail	Yes	Yes	Pass	Successful, No error.

Table 2. Testing summary

3. Conclusion

Although test case ID may not be clear, the test cases are reasonable, and have covered with achieving all requirements. The result is successful.

4 Reference

G. (n.d.). Google Java Style Guide. Retrieved July 26, 2019, from *https://google.github.io/styleguide/javaguide.html*

Ivanov, R. (2019, June 22). Checkstyle Overview. Retrieved July 26, 2019, from *https://checkstyle.sourceforge.io/*

J. (n.d.). Codacy-code review and code quality monitoring. Retrieved August 2, 2019, from *https://www.codacy.com*, *https://app.codacy.com*