CSE 3063 OOSD FALL 2020 - PROJECT EVALUATION PLAN

In order to evaluate your projects at each iteration you need to collect and report detailed statistics about your code. These statistics are listed below. Please use "CSE3063 Fall2020 Project Evaluation Metrics.ods" file to report.

It is crucial for you to provide all the statistics correctly. In your demonstrations we can check these and each mismatch will be penalized by using following formula: Penalty = -10*(|actual value - reported value|)

DOCUMENTS

1- Software Requirement Specification (SRS) document {0}

If you provide a SRS document in this iteration or not

2- Domain Model (DM) {0}

If you provide a DM document in this iteration or not

3- Design UML Class Diagram (DCD) {0}

If you provide a DCD document in this iteration or not

4- Design UML Sequence Diagram (DSD) {0}

If you provide a DSD document in this iteration or not

DESIGN

5- Number of associations {0}

Report how many associations you have between your domain classes in your UML Class Diagram (DCD).

6- Number of one-to-one associations {0}

7- Number of one-to-many associations {0}

8- Number of many-to-many associations {0}

9- Number of aggregations {0}

Report how many aggregation associations you have between your domain classes in your UML Class Diagram (DCD).

10- Number of compositions {0}

Report how many composition associations you have between your domain classes in your UML Class Diagram (DCD).

11- Number of generization (is-a) relationships {0}

Report how many generalization or in other words inheritance relationships in your UML Class Diagram (DCD). If there are several subclasses inheriting from a parent class, count each one of these. In short count all the subclasses.

12- Number of interface implementing classes {0}

Report the number of domain classes that are implementing one or many interfaces in your UML Class Diagram (DCD).

IMPLEMENTATION - CLASSES

13- Number of classes {4}

Number of domain classes in your project in your GitHub repository. Domain classes are the classes created by you to solve the problem. Domain classes do not include user interface or database connection classes. Classes from libraries do not count. Please note that each class should be a separate java file so that you should easily show this in your demos. This is an important grading criteria.

14- Number of abstract classes {0}

Number of abstract classes among your domain classes in your project.

15- Number of interfaces {0}

Number of interfaces among your domain classes in your project.

IMPLEMENTATION - ATTRIBUTES

16- Number of object type attributes {2}

Number of object type attributes your domain classes. Please see [13] for the definition of domain classes. Primitive type attributes such as int double and String attributes do not count. Also public attributes do not count. Only private and protected attributes whose type is one of your domain classes count. This is an important grading criteria.

17- Average number of object type attributes {0.5}

Total object type attributes in your domain classes/number of domain classes. Please see [16] for definition of object type attributes.

18- Minimum number of object type attributes {0}

Among your domain classes find the class with minimum number of object type attributes and report the number of object type attributes in this class.

19- Maximum number of object type attributes {2}

Among your domain classes find the class with maximum number of object type attributes and report the number of object type attributes in this class

20- Rate of object type attributes {0.1}

Object type attributes / total number of attributes including primitives and Strings in your domain classes. Please see [16] for definition of object type attributes. This is an important grading criteria.

IMPLEMENTATION - METHODS

21- Number of methods {10}

Total number of methods in your domain classes.

22- Number of methods with object type input/output parameters {2}

In your domain classes, count only the methods with with object type input/output parameters. This is an important grading criteria.

23- Average number of methods in your domain classes {10}

24- Minimum number of methods in your domain classes {10}

25- Maximum number of methods in your domain classes {10}

26- Rate of methods with object type parameters {0.2}

Total number of methods with object type parameters / total number of methods (including methods with no parameters or with primitive type parameters such as int, Integer, double, Double, etc. and String). Only the methods in your domain classes count. This is an important grading criteria.

IMPLEMENTATION - CODE

27- Total Lines of Code (LOC) {114}

The total lines of functional code you wrote in your domain classes. Empty lines, comments, lines in main function, and library includes do not count.

28- Average class LOC {29}

Average LOC of your domain classes. Please see [27] for how to count lines of code.

29- Minimum class LOC {20}

LOC in your smallest domain class. Please see [18] for how to count lines of code.

30- Maximum class LOC {74}

LOC in your largest domain class. Please see [18] for how to count lines of code.

CODE EXECUTION

31- Proper Execution with standart input {0}

If the code executes without error and produce proper output json file given one of the standart input json file provided in the classroom.

32- Proper Trace and Logging {0}

If the code outputs its execution trace to command line in addition to a log file or not.

33- Proper Execution with extended input {0}

If the code executes without error and produce proper output json file given one of the extended input json file that is not given to the students. This field will be filled by your instructer or TA during your demo.

DIFFERENCE WITH PREVIOUS ITERATION (These will be zero for the first iteration)
34- Lines of code (LOC) added (previous iterations LOC [27] – this iterations LOC [27]) {0}
35- Number of changed classes {0}
36- Number of changed methods {0}
37- Number of newly added domain classes {0}
38- Number of newly added methods {0}
39- Number of newly added attributes {0}
40- Number of deleted classes {0}