**CCDSALG Term 3, AY 2019 – 2020**

**Project 2 Documentation – Calculator (Stack and Queue Application)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Section** | **Names** | **Task 1** | **Task 2** | **Task 3** | **Task 4** | **Task 5** |
| S12 | Gan, John Matthew Ong |  | X | X | X | X |
| S16 | Noblefranca, Jose Noel Cleofe | X | X |  | X | X |
| S15 | Remudaro, Angelo Alvarez |  |  |  | X | X |

Fill this part with your section and names. For the tasks, put an X mark if you have performed the specified task. Please refer to the project specifications for the tasks.

|  |
| --- |
| 1. Programming Language Used: Java |
| 2. Why did you choose the programming language above for Project 2? Explain briefly (1 to 2 sentences).    We chose Java as our programming language because it has a native support for Strings, making it easier to use built-in methods to tokenize expressions. In addition to this, it also has ArrayLists, which make it easy to create data structures for implementing Stacks and Queues, as compared to the limited native libraries of the C language. |
| 3. Depending on the programming language used:   1. List the libraries or APIs that you used in your implementation   Java Standard Libraries   * java.util.regex * java.util.Scanner * java.util.regex.Matcher * java.util.regex.Pattern * java.util.\*  1. Indicate how to compile (if it is a compiled language) your codes, and how to RUN (execute) your program from the COMMAND LINE. Examples are shown below highlighted in yellow. Replace them accordingly. Make sure that all your group members test what you typed below because I will follow them verbatim. I will initially test your solution using the sample input text file that you submitted. Thereafter, I will run it again using my own test data:  * How to compile from the command line (for compiled language only):   C:\CCDSALG>**javac Driver.java**   * How to run from the command line   C:\CCDSALG>**java Driver** |
| 4. How did you implement your data structures (did you use arrays or linked list)? Why? Explain briefly (1 to 2 sentences).  We used an ArrayLists, a kind of dynamic array type, in implementing our data structures, because it is easier to use for storing and accessing data as we would not need to focus too much on the memory aspect. |
| 5. Disclose what is NOT working correctly in your solution. Be honest about this. Explain briefly the reason why your group was not able to make it work. |
| 6. What do you think is the level of difficulty of the project (was it easy, medium or hard)? Which part is hard (if you answered hard)? Type your answer individually for this question.  Gan, John: I would say that the project is of medium difficulty because of the thorough analyzing and trials needed to create and test the code to convert and evaluate the expressions, especially since there were many times where specific test cases would cause the program to fail.  Noblefranca, Jose:  Remudaro, Angelo: |
| 7. Fill-up the table below. Refer to the rubric in the project specs. It is suggested that you do first an individual self-assessment. Thereafter, compute the average evaluation for your group, and encode it below.   |  |  | | --- | --- | | **REQUIREMENT** | **AVE. OF SELF-ASSESSMENT** | | 1. Stack | \_\_\_ (max. 20 points) | | 2. Queue | \_\_\_ (max. 20 points) | | 3. Infix-to-Postfix | \_\_\_ (max. 25 points) | | 4. Postfix Evaluation | \_\_\_ (max. 20 points) | | 5. Documentation | \_\_\_ (max. 10 points) | | 6. Compliance with Instructions | \_\_\_ (max. 5 points) | | **TOTAL SCORE** | \_\_\_ over 100. |   NOTE: The evaluation that the instructor will give is not necessarily going to be the same as what you indicated above. The self-assessment serves primarily as a guide. |