**CMSC203 Assignment 2 Implementation (Documentation)**

Class: CMSC203 CRN 46519

 Program: Assignment #2

Instructor: Farnaz Eivazi

 Summary of Description: Asks user to guess a random number between 0 and 100, giving 7 chances to guess correctly.

 Due Date: 06/30/2022

 Integrity Pledge: I pledge that I have completed the programming assignment independently.

 I have not copied the code from a student or any source.

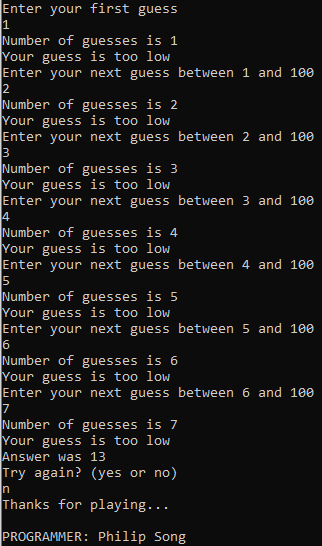
**Part1: Pseudo Code:** Here is a pseudo code for Assignment 2 program:

1. **START**
2. **Generate random number between 0 and 100.**
3. **Tell user to guess the random number between lowest value and highest value**
4. **Get number from user (make sure it is between lowest and highest value)**
   1. **If guess is too low: Update lowest value to current guess**
   2. **If guess is too high: Update highest value to current guess**
   3. **If guess is correct, OR...   
      If user has made 7 guesses total: Go to #6**
5. **Go back to #3**
6. **Ask user if they want to play again**
   1. **If “yes”, go back to #2**
7. **END**

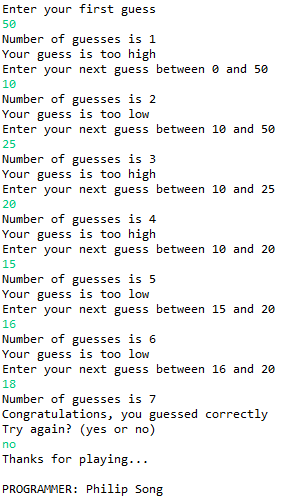
**Part2: Comprehensive Test Plan**

A good test plan should be comprehensive. This means you should have a few test cases that test when the input is in and out of range, division by 0, incorrect Data type, etc. (Provide valid and invalid input)

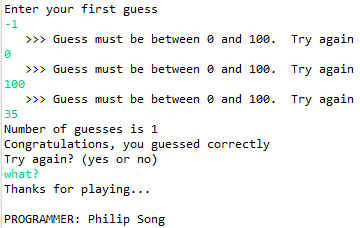
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cases | Input | Expected Output | Actual Output | Did Test Pass? |
| Case 1 | 1  2  3  4  5  6  7  n | Number of guesses is 7  Your guess was too low  Answer was… *13*  Try again? (yes or no)  Thanks for playing… | Number of guesses is 7  Your guess was too low  Answer was… *13*  Try again? (yes or no)  Thanks for playing… | Yes |
| Case 2 | 50  10  25  20  15  16  **18**  no | Numbers of guesses is 7 Congratulations, you guessed correctly Try again? (yes or no)  Thanks for playing… | Numbers of guesses is 7 Congratulations, you guessed correctly Try again? (yes or no)  Thanks for playing… | Yes |
| Case 3 | -1  0  100  **35**  what? | >>> Guess must be between 0 and 100. Try again  >>> Guess must be between 0 and 100. Try again  >>> Guess must be between 0 and 100. Try again  Number of guesses is 1  Congratulations, you guessed correctly Try again? (yes or no)  Thanks for playing… | >>> Guess must be between 0 and 100. Try again  >>> Guess must be between 0 and 100. Try again  >>> Guess must be between 0 and 100. Try again  Number of guesses is 1  Congratulations, you guessed correctly Try again? (yes or no)  Thanks for playing… | Yes |
| Case 4 | 10  20  **16**  yes  **80**  NO | Number of guesses is 3 Congratulations, you guessed correctly Try again? (yes or no)  Enter your first guess  Number of guesses is 1  Congratulations, you guessed correctly  Try again? (yes or no)  Thanks for playing… | Number of guesses is 3 Congratulations, you guessed correctly Try again? (yes or no)  Enter your first guess  Number of guesses is 1  Congratulations, you guessed correctly  Try again? (yes or no)  Thanks for playing… | Yes |

**Part3: Screenshots related to the Test Plan:  
Case 1  
**

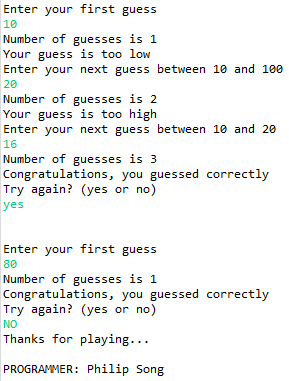
**Case 2**

****

**Case 3**

****

**Case 4**



**Lessons Learned** <Provide answers to the questions listed above>**:**

Write about your Learning Experience, highlighting your lessons learned and learning experience from working on this project.

What have you learned?

* **How to use Java utility class, and without creating objects**

What did you struggle with?

* **I had trouble with inputValidation(): if input was invalid, it would still add +1 to # of guesses**
* **No method to modify count except with resetCount() or through inputValidation()**
  + **I decided to ignore the issue by declaring a separate int count variable to keep track of # of guesses**

What would you do differently on your next project?

* **If I were allowed to change RNG.java, I would have added an else statement to inputValidation so it would not count invalid guesses**

What parts of this assignment were you successful with, and what parts (if any) were you not successful with?

* **Recreating sample output (close enough, because randomness)**
* **Working around the inputValidation() issue**

Provide any additional resources/links/videos you used to while working on this assignment/project.

**Check List:** <Provide answers to the column Y/N or N/A >**:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment#\_Moss.zip | **Yes** | **No MOSS** |
|  | * FirstInitialLastName\_Assignment#.docx/.pdf | **Yes** |  |
|  | * Source java files | **Yes** |  |
|  | **Program compiles** | **Yes** |  |
|  | **Program runs with desired outputs related to a Test Plan** | **Yes** |  |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **Yes** |  |
|  | * Screenshots related to the Test Plan | **Yes** |  |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) | **Yes** |  |
|  | * UML Diagram (if required) | **Yes or No or N/A** |  |
|  | * Algorithms/Pseudocode (if required) | **Yes** |  |
|  | * Flowchart (if required) | **Yes or No or N/A** |  |
|  | * Lessons Learned | **Yes** |  |
|  | * Checklist is completed and included in the Documentation | **Yes** |  |