**CMSC203 Assignment 2 Implementation (Documentation)**

Class: CMSC203 CRN 38511

Program: Assignment #2

Instructor: Prof. Grinberg

Summary of Description: Build an application that will receive a guess and report if your guess is the random number that was generated.

Due Date: 02/26/2022

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. **Driver Class – RandomNumberGuesser**
2. **Inside main method:**
3. **Create a Scanner object**
4. **Declare int variables randNum, nextGuess, highGuess, lowGuess.**
5. **Declare String variable yesNo.**
6. **Print header.**
7. **Inside while loop that repeats the guessing game until the “yesNo” variable = “no”.**
8. **Initialize variables: highGuess = 100, lowGuess = 1, randNum = the random number generated in the “rand” method**
9. **Call the method resetCount to reset any existing count numbers**
10. **Print text asking the user for an initial guess of the random number between 0 and 100.**
11. **Read user input and stores in int variable nextGuess**
12. **Inside nested while loop that continues the guessing game until the user guesses correctly**
13. **Call inputValidation() method to check if the number entered is valid**
14. **If the number is not valid then loops back to asking the user to enter a valid number.**
15. **If the number is valid, then end the input validation method and continue**
16. **Print out the number of guesses using the getCount() method**
17. **Print if the number guessed is too high or too low.**
18. **Print out the range of the random number**
19. **Print text asking for another guess**
20. **Read user input and stores in num variable**
21. **If nextGuess is equal to randNum then print “your guess is correct”**
22. **Print text asking the user if they would like to play again**
23. **Read user input**
24. **if user input yesNo = no, then end the first while loop, otherwise continue the first while loop**
25. **closing statement**
26. **terminate the program**

**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 | 50  25  40  45  47  46  Yes  75  25  20  15  17  no | # of Guesses = 1  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4  # of Guesses = 5  # of Guesses = 6 correct  # of Guesses = 1  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4  # of Guesses = 5 correct  Thanks for playing | # of Guesses = 1  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4  # of Guesses = 5  # of Guesses = 6 correct  # of Guesses = 1  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4  # of Guesses = 5 correct  Thanks for playing | yes |
| Case 2 | 1000  35  20  10  15  14  15  12  13  no | Error (try again)  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4  # of Guesses = 5  # of Guesses = 6  Error (try again)  # of Guesses = 8  # of Guesses = 9 correct  Thanks for playing | Error (try again)  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4  # of Guesses = 5  # of Guesses = 6  Error (try again)  # of Guesses = 8  # of Guesses = 9 correct  Thanks for playing | yes |
| Case 3 | -5  1  2  50  no | Error (try again)  Error (try again)  # of Guesses = 3  # of Guesses = 4 correct  Thanks for playing | Error (try again)  Error (try again)  # of Guesses = 3  # of Guesses = 4 correct  Thanks for playing | yes |
| Case 4 | 100  50  30  40  no | Error (try again)  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4 correct  Thanks for playing | Error (try again)  # of Guesses = 2  # of Guesses = 3  # of Guesses = 4 correct  Thanks for playing | yes |

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

**Case 1**

**Graphical user interface, text, application, email

Description automatically generated**

**Case 2**

**Graphical user interface, table

Description automatically generated with medium confidence**

**Case 3**

**Graphical user interface, text, application, email

Description automatically generated**

**Case 4**

**Graphical user interface, text, application, email

Description automatically generated**

**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?

I have gained more practice using nested loops, I have learned to pass data between 2 classes

What did you struggle with?

I struggled with the while loops, I ended up spending way too much time trying to figure out the best way to write this program with loops. I had a little trouble trying to pass data from both classes, but I was able to watch a quick YouTube video explaining the important thing to use.

What would you do differently on your next project?

I would add methods for the class, it would make reading my source code much eisierd for less confusion

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

I was successful with every part

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** |  |
|  | * 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) | **N/A** |  |
|  | * UML Diagram (if required) | **N/A** |  |
|  | * Algorithms/Pseudocode (if required) | **Yes** |  |
|  | * Flowchart (if required) | **N/A** |  |
|  | * Lessons Learned | **Yes** |  |
|  | * Checklist is completed and included in the Documentation | **Yes** |  |