Project PC21 Ch5 Write-Up

**Meena Chandok**

**Pseudo-code**

1. Included <cstdlib> for rand, srand and <ctime> for time function.
2. Variable declaration and Value assignment for variables. Computer generated Random Number (Num), User’s guess Answer (Ans) and Number of Attempts (NumAttempt) by user to guess the number correctly declared as integer. Set the default value of Number of Attempts as: NumAttempt=0 and unsigned seed = time (0) (to generate the random number whenever program runs). Declared srand(seed) to seed the random number generator.
3. Display the purpose of the program.
4. One can use any range to generate the random numbers. In this program the random number generation set in the range of 1-100.
5. Ask the user if they can guess the random number generated by computer and give the user option to enter their guess number/ answer.
6. The input validation will be not to accept any number outside the range, including any character, string or symbol from user. For this the step is designed to make the program quit/exit if user’s input is in the form of any character, string or symbol.
7. Once the user provide input in the form of answer, the while loop of the program will check for the conditions ((Num != Ans) && cin.good()). If the condition check appears to be false (NO) then program will further evaluate the conditions for cin.good. If cin.good appears to be false then the program will exit by declaring as an “invalid input”. For the condition of cin.good as true and (Num==Ans), the message of congratulating user for guessing the number correctly and displaying the total number of attempts taken to guess the number will be displayed. (Num==Ans), will be a strict check as no character, string or symbol input by user can be equal to Num , therefore when met this condition the message of congrats will be displayed.
8. If the user’s input in #7 is ((Num != Ans) && (cin.good)) is true then the program will perform next validation to check the range of numbers. Since the range of random number generation in this program is designed to be in the range of 1-100, the input validation will be not to accept any –ve number, zero (0) or number greater than 100. If user enters any negative number, 0 or the number outside the range of 1-100 then the program will display the message that the number should be in range of 1-100 and provides the user another option to enter the number again. The program is set not to count these as Number of Attempts in guessing the number correctly, as user did not enter the valid range. The count of Number of Attempts will start once user start putting a number in valid acceptable range. Until user enters a valid input the program won’t proceed further and any invalid range input will not be counted towards Number of Attempts.
9. Once user’s the input answer is in acceptable range the “else” with nested “if” conditions will be executed to check if Num<Ans and Num>Ans as follows:
   1. Condition 1: Num<Ans - If user’s answer number is higher than the generated random number then display the message that “number is higher” and provides user an option to enter the number again. Any guess attempt of user will be counted towards the number of attempts he/she made to guess the number correctly.
   2. Condition 2: Num>Ans -If user’s answer number is lower than the generated random number then display the message that “number is lower” and provides user an option to enter the number again. Any guess attempt of user will be counted towards the number of attempts he/she made to guess the number correctly.
10. Once (Num == Ans) the loop in #9 will exist by displaying the message congratulating user for guessing the number correctly and display the total number of attempts taken to guess the number.
11. Steps 5-10 are also shown in the form of flow chart below.



**Test Cases:**

**Valid Data** when asking for user to guess the computer generated random number:

User Answer: Ans

Ans: Integer in range of 1-100 (positive values only)

**Invalid Data** for user input:

User Answer: Anything other than the integer in range of 1-100.

Any character, symbol and string etc.

Negative values, and any positive value greater than the 100.

**Test Case Artifacts:**

|  |  |  |
| --- | --- | --- |
| **Token Input when asked For**  **Guessing the number** | **Expected Output** | **Actual Output** |
| Guessing the number | "Can you Guess the number?"  "Enter your answer" | "Can you Guess the number?"  "Enter your answer" |
|  |  |  |
| **Possible Input for Number** | **Expected Output** | **Actual Output** |
| Any character, string, symbol | "invalid entry" | "invalid entry" |
| e.g. =, @, yes, no, Han, +,  name | "invalid entry" | "invalid entry" |
| Negative numbers  -1,-43,-67888 etc. | "The number should be in range of 1-100. Please enter again" | "The number should be in range of 1-100. Please enter again" |
| Positive numbers >100  e.g. 101, 173, 6543, 98765 etc. | "The number should be in range of 1-100. Please enter again" | "The number should be in range of 1-100. Please enter again" |
| Positive number in range of 1-100.  e.g. 34 (in this case when tested) | "Your number is lower than the random number, please enter another number" | "Your number is lower than the random number, please enter another number" |
| Positive number in range of 1-100.  e.g. 78 (in this case when tested) | "Your number is higher than the random number, please enter another number" | "Your number is higher than the random number, please enter another number" |
| Positive number in range of 1-100.  e.g. 45 (in this case when tested) | "Your number is higher than the random number, please enter another number" | "Your number is higher than the random number, please enter another number" |
| Positive number in range of 1-100.  e.g. 40 (in this case when tested) | "Congrats!! You guessed the correct number " | "Congrats!! You guessed the correct number " |
| **Expected outcomes For**  **Number of Attempts** | **Expected Output** | **Actual Output** |
| Positive number in range of 1-100.  e.g. 40 (in this case, when tested) as part of game trial shown above. | "You have taken a total of 4 attempts to guess the number correctly" | "You have taken a total of 4 attempts to guess the number correctly" |
|  |  |  |

Assumptions:

I am assuming that the user will follow instructions for guessing the random number.

Learning Experience:

I believe I did fairly well on this project. I got a better appreciation of the loop and use of else and if statements in loop.

I learned how to use ‘avoid’ the program to crash if user enters the character, symbol or any other string value by using cin.good phrase.

I also tried the same program with do-while loop which performed equally good. It would be interesting to generate the random numbers in higher digits and double to see if the program still performs equally good.