Assignment 1

Objective:

The objective of this assignment is to create a simple game. This game will help you practice the use of if statements, while loops, switch statements, and functions in C++.

Requirements:

Create a player vs computer game, the game begins with a pile of 20 sticks. The player/computer take turns withdrawing 1, 2, or 3 sticks at a time. Whoever withdraw the last stick loses the game.

1. **Game Flow**:
    - The game should display a menu first, and the menu can look like this:

      Welcome to the Game!

      1. Play Game

      2. Show Rules

      3. Exit
    - If 2 is chosen, game's rules should be displayed; If 3 is chosen, the game will exit; Otherwise the following flow should be followed.
    - The program should generate a random number to determine who should start the game.
    - When it is the player's turn, the program should perform input validation to make sure that the entered number is between 1 and 3.
    - When it is the computer's turn, it should play according to the following rules:
        1. If there are more than 4 sticks left, the computer should randomly generate the number (1-3).
        2. If there are 2 to 4 sticks left, then the computer should withdraw enough sticks to leave 1.
        3. If there is 1 stick left, then the computer must take it and loses.
    - After a winner is determined, announce the winner and ask whether the game should be played again.
2. **User Menu**:

- Implement a user menu using a switch statement to handle the player's choices:
  - Option 1: Play the game.
  - Option 2: Show the rules.
  - Option 3: Exit the game.

3. **Functions**:
   - Create a function void displayRules() that displays the game rules.
   - Create a function void playGame() that contains the game logic.
   - Create a function int getComputerNumber() that generates and returns a random number between 1 and 3.
   - Create a function int userInput() to validate the player enters a valid number (1, 2, or 3)
   - Create a function bool askToPlayAgain() that asks the player if they want to play again and returns true if they do, false otherwise.
4. **Loops**:
   - Use a while loop to allow the player to keep playing until a winner is found.
   - Use another while loop to continue playing the game if the player chooses to play again.

Grading Criteria:

- Correct use of functions, if statements, loops, and switch statements.
- Proper handling of user input and input validation.
- Clean and readable code with comments explaining the logic.
- Successful implementation of game logic and replay functionality.

Submission:

Submit the C++ source file (.cpp) and screen shot of your program output on Moodle. Ensure that your code compiles and runs correctly.