

Lab 7

Due on: November 8th, Friday 11:59 pm

Task: Create skeleton codes with function stubs for Assignment 9

For Assignment 9, we make the advanced version of Task 2 in Assignment 8. The objective is to create a C++ based game in which one player (the user) competes against the computer where each player tries to steal the points from the opponent. Each player starts out with 100 points. The game is won by the first player to reach a set number of points. The number of points required to win the game is to be inputted by the user. Each turn, the player (human or computer) repeatedly rolls a dice until either a 3 is rolled or the player chooses to hold. If the player chooses to hold, the sum of the rolls (i.e., the *turn total*) is calculated. At any time during a player's turn, the player is faced with a binary decision:

- **roll [0]:** If the player rolls a
 - **3:** the player gives 3 points to the opponent and it becomes the opponent's turn. (The *turn total* is -3.)
 - **1, 2, 4, 5, 6:** the number is added to the player's turn total and the player's turn continues (i.e., gets to roll again).
- **hold [1]:** The *turn total* is calculated. The *turn total* is subtracted from the opponent's score and added to the current player's score. It becomes the opponent's turn.

Check whether a winner has been established at the end of every turn. The player who takes the first turn in the game is chosen randomly.

Do NOT use: global variables, arrays, pass by reference functions, or pass by pointer functions.

Requirements:

1. Your program should contain the following three functions (besides the main function, of course):

roll: takes in nothing. Returns a random integer in {1, 2, 3, 4, 5, 6}

oneTurn: takes in a single boolean variable to indicate whether it's user's or computer's turn (e.g., 1 = human turn, 0 = computer turn). Plays out the entire turn until player either rolls a 3 or chooses to hold. Returns the turn total as an integer value. (*Note: this function should be calling the roll function*). The compute player should:

- Roll until it reaches at least some minimum turn total (You can decide the minimum turn total)
- Have some randomness in its decision making
- Be capable of rolling or holding past the minimum

loopGame: takes in a single integer value for game points goal (i.e., the number of points player needs to reach to win the game). Continues to repeat until either computer player or human player has won. Returns a boolean value (1 = computer win, 0 = human win). (*Note: this function should be calling the oneTurn function*).

2. Your program should interactively display the player and the computer choices, the dice rolls, the running turn total and the running score after each turn.

3. You will need to use randomization for this assignment. Make sure you are seeding the randomizer so your program doesn't end up generating the same pseudorandom sequence every time.
4. For interacting with the user, you must use `cin` and `cout`.

Instructions to submit your Lab work

For the lab assignment, come up with the algorithm for the project. Create skeleton codes with function stubs. Figure out what variables the functions take in, what they return, and what variables you will most likely need. Your comments should contain the pseudocode describing the logic of your program. Make sure your skeleton code compiles. Submit your .cpp file on Moodle under Lab 7 by the due date.

Keep in mind the Honor code and ensure that you do not violate any of the rules it entails.