CptS 121 - Program Design and Development

Programming Assignment 4: A Game of Chance "Craps"

Assigned: Wednesday, September 25, 2024
Due: Wednesday, October 9, 2024 by midnight

I. Learner Objectives:

At the conclusion of this programming assignment, participants should be able to:

- Apply repetition structures within an algorithm
- Construct while (), for (), or do-while () loops in C
- Compose C programs consisting of sequential, conditional, and iterative statements
- Eliminate redundancy within a program by applying loops and functions
- Create structure charts for a given problem
- Determine an appropriate functional decomposition or top-down design from a structure chart
- Generate random numbers for use within a C program

II. Prerequisites:

Before starting this programming assignment, participants should be able to:

- Analyze a basic set of requirements and apply top-down design principles for a problem
- Customize and define C functions
- Apply the 3 file format: 1 header file and 2 source files
- Open and close files
- Read, write to, and update files
- Manipulate file handles
- Apply standard library functions: fopen (), fclose (), fscanf (), and fprintf ()
- Compose decision statements ("if" conditional statements)
- Create and utilize compound conditions
- Summarize topics from Hanly & Koffman Chapter 4 & 5 including:
    - What are counting, conditional, sentinel-controlled, flag-controlled, and end file-controlled loops
    - What are while (), do-while (), and for () loops
    - What is a selection or conditional statement
    - What is a compound condition
    - What is a Boolean expression
    - What is a flowchart

III. Overview & Requirements:

The following description has been adopted from Deitel & Deitel. One of the most popular games of chance is a dice game called "craps," which is played in casinos and back alleys throughout the world. The rules of the game are straightforward:

A player rolls two dice. Each die has six faces. These faces contain 1, 2, 3, 4, 5, and 6 spots. After the dice have come to rest, the sum of the spots on the two upward faces is calculated. If the sum is 7 or 11 on the first throw, the player wins. If the sum is 2, 3, or 12 on the first throw (called "craps"), the player loses (i.e. the "house" wins). If the sum is 4, 5, 6, 8, 9, or 10 on the first throw, then the sum becomes the player's "point." To win, you must continue rolling the dice until you "make your point." The player loses by rolling a 7 before making the point.

Write a program that implements a craps game according to the above rules. The game should allow for wagering. This means that you need to prompt that user for an initial bank balance from which wagers will be added or subtracted. Before each roll prompt the user for a wager. Once a game is lost or won, the bank balance should be adjusted. As the game progresses, print various messages to create some "chatter" such as, "Sorry, you busted!", or "Oh, you're going for broke, huh?", or "Aw cmon, take a chance!", or "You're up big, now's the time to cash in your chips!"

Use the below functions to help you get started! You may define more than the ones suggested or define all of your own functions if you wish!

- (5 pts) void print_game_rules (void) — Prints out the rules of the game of "craps".
- (5 pts) double get_bank_balance (void) - Prompts the player for an initial bank balance from which wagering will be added or subtracted. The player entered bank balance (in dollars, i.e. $100.00) is returned.
- (5 pts) double get_wager_amount (void) - Prompts the player for a wager on a particular roll. The wager is returned.
- (5 pts) int check_wager_amount (double wager, double balance) - Checks to see if the wager is within the limits of the player's available balance. If the wager exceeds the player's allowable balance, then 0 is returned; otherwise 1 is returned.
- (5 pts) int roll_die (void) - Rolls one die. This function should randomly generate a value between 1 and 6, inclusively. Returns the value of the die.
- (5 pts) int calculate_sum_dice (int die1_value, int die2_value) - Sums together the values of the two dice and returns the result. Note: this result may become the player's point in future rolls.
- (10 pts) int is_win_loss_or_point (int sum_dice) - Determines the result of the *first* dice roll. If the sum is 7 or 11 on the roll, the player wins and 1 is returned. If the sum is 2, 3, or 12 on the first throw (called "craps"), the player loses (i.e. the "house" wins) and 0 is returned. If the sum is 4, 5, 6, 8, 9, or 10 on the first throw, then the sum becomes the player's "point" and -1 is returned.
- (10 pts) int is_point_loss_or_neither (int sum_dice, int point_value) - Determines the result of any successive roll after the first roll. If the sum of the roll is the point_value, then 1 is returned. If the sum of the roll is a 7, then 0 is returned. Otherwise, -1 is returned.
- (5 pts) double adjust_bank_balance (double bank_balance, double wager_amount, int add_or_subtract) - If add_or_subtract is 1, then the wager

amount is added to the bank_balance. If add_or_subtract is 0, then the wager amount is subtracted from the bank_balance. Otherwise, the bank_balance remains the same. The bank_balance result is returned.

- (5 pts) void chatter_messages (int number_rolls, int win_loss_neither, double initial_bank_balance, double current_bank_balance) - Prints an appropriate message dependent on the number of rolls taken so far by the player, the current balance, and whether or not the player just won his roll. The parameter win_loss_neither indicates the result of the previous roll.
- (10 pts) Others?
- (20 pts) A main ( ) function that makes use of the above functions in order to play the game of craps as explained above. Note that you will most likely have a loop in your main ( ) function (or you could have another function that loops through the game play).

Have a great time with this assignment! There is plenty of room for creativity! Note: I have not stated how you must display the game play! You may do as you wish!

IV. Submitting Assignments:

1. Using Canvas https://canvas.wsu.edu/, please submit your solution to the correct "Programming Assignments" (PA) folder. Your solution should be zipped into a .zip file with the name `<your last name>_PA4.zip` and uploaded. To upload your solution, please navigate to your correct Canvas ***lab*** course space. Select the "Assignments" link in the main left menu bar. Navigate to the correct PA submission folder. Click the "Start Assignment" button. Click the "Upload File" button. Choose the appropriate .zip file with your solution. Finally, click the "Submit Assignment" button.
2. Your .zip file should contain your one header file (a .h file), two C source files (which must be .c files), and project workspace. Delete the debug folders before you zip the project folder.
3. Your project must build properly. The most points an assignment can receive if it does not build properly is 65 out of 100.

V. Grading Guidelines:

This assignment is worth 100 points. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

- 90 pts for adherence to functional decomposition stated above (see the individual points above)
- 10 pts for adherence to proper programming style established for the class and comments