CptS 121 – Program Design and Development

Programming Assignment 7: Poker (5-Card Draw)

Assigned: Friday, November 8, 2024
Due: Friday, November 22, 2024 (extended due date)

I. Learner Objectives:

At the conclusion of this programming assignment, participants should be able to:
- Declare and define arrays of pointers to strings
- Implement and apply structs
- Manipulate 2-Dimensional arrays
- Apply triply nested loops
- Manipulate strings

II. Prerequisites:

Before starting this programming assignment, participants should be able to:
- Apply and implement pointers in C
- Pass output parameters to functions
- Analyze a basic set of requirements and apply top-down design principles for a problem
- 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
- Open and close files
- Read, write to, and update files
- Manipulate file handles
- Apply standard library functions: fopen (), fclose (), fscanf (), and fprintf ()
- Apply and implement pointers 2-dimenional arrays
- Define and apply structs in C
- Define and apply strings in C
- Compose decision statements ("if" conditional statements)
- Create and utilize compound conditions
- Summarize topics from Hanly & Koffman Chapter 8 including:
  - What is an array?
  - Distinguishing between single dimensional and 2-dimentional arrays
  - What is an index?

III. Overview & Requirements:

Write a program that allows a user to play 5-Card-Draw Poker against the computer.

You must start with the following example code supplied by Deitel & Deitel .This will help you get started with the game of Poker. Please read this site to learn the rules of Poker http://en.wikipedia.org/wiki/5_card_draw. Complete the following steps and you will have a working Poker game!!!

Adapted from Deitel & Deitel's C How to Program (6th Edition):
(1) In order to complete the game of 5-card-draw poker, you should complete the following:
   (a) (5 pts) Declare a struct called Card that contains two integers. One integer represents the index of where to find the face value of the card in the array of strings, and the other integer represents the index of where to find the suit of the card in the other array of strings. Recall, these arrays are already supplied in the example code.
   (b) (5 pts) Declare a struct called Hand that contains an array of 5 struct Cards.
   (c) (5 pts) Create a menu to that allows the user to view the game rules, play the game, and exit the game.
   (d) (10 pts) Modify the card dealing function provided in the example code so that a poker hand is dealt. You should be able to use this function to deal and draw any number of cards. However, it is fine if you decide to implement a separate draw () function.
   (e) (5 pts) Write a function to determine if the hand contains a pair.
   (f) (5 pts) Write a function to determine if the hand contains two pairs.
   (g) (5 pts) Write a function to determine if the hand contains three of a kind (i.e. three jacks).
   (h) (5 pts) Write a function to determine if the hand contains four of a kind (i.e. four aces).
   (i) (5 pts) Write a function to determine if the hand contains a full house (i.e three of a kind and two of another).
   (j) (5 pts) Write a function to determine if the hand contains a flush (i.e. all five cards of the same suit).
   (k) (5 pts) Write a function to determine if the hand contains a straight (i.e. five cards of consecutive face values).

(2) (15 pts) Simulate the dealer. The dealer's five-card hand is dealt "face down" so the player cannot see it. The program should then evaluate the dealer's hand, and based on the quality of the hand, the dealer should draw one, two, or three more cards to replace the corresponding number of unneeded cards in the original hand. The program should then re-evaluate the dealer's hand.

(3) (15 pts) Make the program handle the dealer's five-card hand automatically. The player should be allowed to decide which cards of the player's hand to replace. The player can replace up to three cards. The program should then evaluate both hands and determine who wins. The game should be played until the user wants to exit.

You may make any adjustments or customizations to your Poker game that you wish!!!
Have fun with this assignment!!!

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>_PA7.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 project must contain one header file (a .h file), two C source files (which must be .c files), and project workspace.
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:

1. (90 pts) for adherence to the above requirements (please see the individual point totals above)
2. (10 pts) for "good" style and design (i.e. proper function declarations, definitions, and comments)