**Pseudo Code**

// Program Open

// Add libraries, namespace and headers

// Start Main

// Set random seed

// Declare variables

// Open wins file

// Read in wins & total

// Close wins file

// Check if player has played before

// Make sure we don't divide by 0!

// If total !0

// Calculate win/loss ratio

// Output current win/loss ratio

// Else Introduce game, get player name

// Assign face values to the cards to match deck

// Check game score (GAME LOOP BEGINS) do while scores < 50

// initialize indx for linear search

// Set initial trick scores to 0 & match to false

// Shuffle deck & face values

// Deal Cards - 13 to each player, in order shuffled

// Sort player hands

// Bubble sort player

// Selection sort Larry

// Mark sort Curly & Moe

// Begin loop to play the hand (13 times - 1 per card in hand)

// reset index to 0

// If first trick find who has 2 clubs

//perform linear search to find 2 clubs

// While loop to go through the deck

// Increments the indx if not found

// ends the loop when 2 clubs selected

// return the indx 2 clubs was at

// assign initial player based on who has 2 clubs

// If indx is 0-12 player first, then larry, curly, moe

// If indx is 13-25 larry first, then curly, moe, player

// if indx is 26-38 curly first, then moe, player, larry

// else (indx 39-51) moe first, then player, larry, curly

// set match to false for all

// Begin loop to play trick (4 times - 1 per player)

// If player's turn (order = trick)

// Print cards

// Get player choice - do/while validate card is in range and has not been played

// If player has 2 clubs, force that play

// set player match to true

// Else if player not first, enforce suit compliance

// if player matched suit set to true, otherwise it stays false

// Else (player is first in the trick) player picks any card

// set match to true

// Print out what the player played

// Set the player's choice to = value of the hand[] they played for scoring

// If Stooge's turn (check to see if their count = trick)

// PlayCard program begins

// Set chosen to false (when choice is set chosen = true returns to main)

// Determine if Stooge order = 0

// If Stooge has 2 clubs, force that play

// Stooge first but no 2 clubs

// play smallest card

// If Player order = 0

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Else If Larry order = 0 (& Stooge is not Larry)

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Else If Curly order = 0 (& Stooge is not Curly)

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Else Moe order = 0

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Output the card Played

// Set the Stooge's choice to = value of the hand[] they played for scoring

// Score the trick

// all played hearts trick worth 4pts, largest card wins

//\*\*\*\*\*\*\*\*\*\*\*\*\* (process below is the same for 3 hearts, 2 hearts, no hearts)

// determine how many hearts were played

// find who matched the initial suit and had the highest card of that suit

// check if Q spades played

// 3 hearts: 16 pts, 2 hearts: 15 pts, 1 heart: 14 pts, no hearts: 13 pts

// output winner & assign trick score

// no Q spades played

// 3 hearts: 3pts, 2 hearts: 2 pts, 1 heart: 1 pt, 0 hearts: 0 pts

// output winner & assign trick score

// unset player choice from card value to 1-13

// increment hand with count();

// unset each Stooge choice from card value to 1-13

// end trick (trick increments)

// hand ends when hand has been played 13 times

// shuffle deck

// deal

// resort everyone's cards

// If anyone shoot moon - assign 26 points to all other opponents set that person's hand to 0

// Else add the hand score to the game score

// Output game score

// check if player wants to continue (do while to force choice to be y or no)

// if N then exit program

// check if player won & output result

// calculate player winLoss

// output winLoss %

// open wins, output wins & total, close the file

// END PROGRAM