|  |  |
| --- | --- |
| **Function Descriptions and Comments** | |
| **Function Name** | **Description** |
| void banner(string) | Parameters: string for the filename  Displays banner character image from a file |
| void filName(vector<string> crdName, ifstream &names) | Parameters: vector<string> crdName is an empty vector from plyGame(); ifstream &names is a files with string names  Uses names file to fill the crdName vector |
| void filIds(short []crdID, ifstream &ids,const short DK\_SIZE) | Parameters: empty short array sized to DK\_SIZE, ifstream object connected to a file of short ints; const short DK\_SIZE is the size of the array  Uses the referenced file to fill the crdID array with values |
| void filVals(short []crdVal, ifstream &vals,const short DK\_SIZE) | Parameters: empty short array sized to DK\_SIZE, ifstream object connected to a file of short ints; const short DK\_SIZE is the size of the array  Uses the referenced file to fill the crdVal array with values |
| void filSuit(short []crdSuit, ifstream &suits,const short DK\_SIZE) | Parameters: empty short array sized to DK\_SIZE, ifstream object connected to a file of short ints; const short DK\_SIZE is the size of the array  Uses the referenced file to fill the crdSuit array with values |
| void plyGame() | Main function for gameplay |
| char GtDealr() | Uses rand() to generate a character which will decide who gets dealt first—the computer or the Player. ‘C’ for Comp; ‘P’ for Player. The one who gets dealt first also gets “nibs”. This information will be outputted as a message.  Output: A char with ‘P’ for player and ‘C’ for computer |
| void deal(short deck[], vector<short> &handC, vector<short> &handP ,char dealer) | Parameters: deck[],vector<short> &handC, vector<short> &handP, and dealer from plyGame().  Uses and changes the deck array, the computer’s hand vector, and the player’s hand vector using addCard(). Char dealer contains ‘P’ for player or ‘C’ for computer. Either starts dealing to the computer or Player based on the value of dealer. Copies the value to a hand, and then changes it in the deck to -1 to indicate that it was used. Loops for six cards to each player. |
| short addCard(short deck[], vector<short> &) | Parameters: deck[] from plyGame(); vector<short> is an empty vector  Uses and the deck array. Changes the computer’s hand vector, and the player’s hand vector. Uses SlCrdDk() for array sub number to select. Uses inDeck() to determine if card is available.  Output: card ID between 0-51 |
| short SlCrdDk() | Uses rand() to select a number between 0 and 51, that value is returned as a short. The number is used for an array sub number |
| bool inDeck(short cardSub, const short[]) | Parameters: cardSub is a short between 0-51 and the short[] is the deck array from plyGame().  Uses cardSub to determine if the card is available for selection or has already been selected  Output: boolean, true if card is available; otherwise false |
| bool inHand(short id, const vector<short> &) | Parameters: cardSub is a short between 0-51 and the vector is a “hand” with card ID’s  Uses id to search through the vector to determine if the card is available for selection  Output: boolean, true if card is available; otherwise false |
| bool inHndC(short sub, const vector<short> &hand) | Parameters: sub is a short between 0-51 and the vector is the computer’s hand from plyGame().  Uses sub to determine if the card is available for selection or has already been selected. Used for computer’s hand only.  Output: boolean, true if card is available; otherwise false |
| short cardID(string, const vector<string> &crdName, const short crdID[], const short DK\_SIZE) | Parameters: string is a card name; const string crdName[] from plyGame(); const short crdID[] from plyGame(); const short DK\_SIZE for looping through array  First validates that the string name is a string in the array. If yes, it returns the card ID value in the corresponding sub position of crdID[]. If no, returns -1 to indicate that the card name does not exist  Output: short card ID between 0-51 or if invalid card name -1 |
| void dspHand(vector<short> handP , vector<string> crdName) | Parameters: vector handP is the player’s hand containing card ID’s; vector<string> crdName from plyGame() is a vector containing string names for cards.  Uses the values from the handP vector to look up values in the crdName[] array and displays each card ID which is > 0 |
| short strtCrd(short deck[],bool &isStJak) | Parameters: uses deck[] from plyGame() which holds available card values; uses reference to boolean isStJak from plyGame().  Uses SlCrdDk() to get a random value. Validates that the corresponding card is available using inDeck(). Assigns a value of -1 in the deck to indicate that it was used. Checks if the card equals values for Jacks, if so isStJak=True, else false.  Output: Returns a short containing a card ID |
| void banner(char dealer, string) | Parameters: bool dealer from plyGame(); string for the filename  !OVERLOADED FUNCTION! Character dealer is used to indicate who gets nibs. Displays a banner from a file. Displays a message indicating that the bonus will be applied to the dealer |
| void dspHand(short, vector<string> crdName) | Parameters: the short is a card ID from 0-51; vector<string> crdName from plyGame() is a vector containing string names for cards.  !OVERLOADED FUNCTION! Uses the short to display the name for the card ID. |
| void rmveCrd(short, const vector<short> &) | Parameters: the short is a card ID 0-51; vector<short> is any vector being used as a hand with card values  Searches through the vector for the card ID and changes it to -1, effectively removing the card from the hand. |
| void discrdP(vector<short> &handP,vector<short> &crib, const vector<string> crdName , const short cardID[]) | Parameters: vector<short> &handP is Player’s hand from plyGame(); vector<short> &crib from plyGame();const vector<string> crdName from plyGame() used for dspHand() and cardID(); const short cardID[] from plyGame() used for cardID()  Loops twice. Displays Player’s hand using dspHand(). Gets a card name from Player, validates and gets card ID from cardID(). Validates that card is in Player’s hand using inHand(). Assigns card ID to vector<short> &crib. Changes card ID to -1 in Player’s hand using rmveCrd(). |
| void discrdC(vector<short> &handC,vector<short> &crib, const cardID[]) | Parameters: vector<short> &handC is computer’s hand from plyGame(); vector<short> &crib is crib hand from plyGame(); const cardID[] is card ID’s from plyGame()  Loops twice. Randomly selects a card to discard to crib hand. Uses rand()%6 to select a value 0-5 to use as a sub number. Validates that sub number has a valid card ID using inHand(). Assigns card ID to vector<short> &crib. Changes card ID to -1 in computer’s hand. |
| void copyVec(vector<short> &, vector<short> &) | Parameters: two vectors of the same datatype  Takes the values in the first vector and copies them to the second vector. |
| short cardVal(short, const short crdVal[]) | Parameters: a short int card ID from 0-51; crdVal[] is the array of card values from plyGame()  Uses the card ID to determine the card’s value  Output: a short with the value 1-10 |
| void plyCrdC(vector<short> &c\_handC, const short crdVal[], const vector<string> &crdName, unsigned short &playCnt, bool &cmpPass, int &pointsC, char &lastPly) | Parameters: vector<short> &c\_handC is a copy of the computer’s hand from plyGame(); const crdVal[] is an array of card values from plyGame() for cardVal(); const vector<string> crdName[] is an vector of card names for dspHand() from plyGame(); unsigned short &playCnt has value 0-31 from plyGame(); bool &cmpPass initialized to false from plyGame(); int &pointsC is points total from plyGame(); char &lastPly holds ‘C’ or ‘P’ from plyGame()  Used to have the computer play a card from its “hand”. Loops through the hand looking for a card ID >=0. Uses cardVal() to get the card’s value. Checks if card’s value plus playCnt > 31. Returns cmpPass=True if loop ends without selecting a card. If cmpPass=False—displays card using dspHand(), assigns -1 to card ID in hand, calculates playCnt, and sets lastPly = ‘C’. Displays playCnt. |
| void plyCrdP(vector<short> &c\_handP, const short crdID[], const short crdVal[], const vector<string> &crdName, unsigned short &playCnt, bool &plrPass, int &pointsP, char &lastPly) | Parameters: vector<short> &c\_handP is a copy of the Player’s hand from plyGame(); const short crdID[] is an array of card id’s for cardID() from plyGame(); const crdVal[] is an array of card values from plyGame() for cardVal();const vector<string> crdName is a vector of card names for dspHand() from plyGame(); unsigned short &playCnt has value 0-31 from plyGame(); bool &plrPass initialized to false from plyGame(); int &pointsP is points total from plyGame(); char &lastPly holds ‘C’ or ‘P’ from plyGame()  Used to have Player play a card from their hand. Displays Player’s hand using dspHand(). Gets a card name from Player, validates and gets card ID from cardID(). Validates that card is in Player’s hand using inHand().Uses cardVal() to get the card’s value. Checks if card’s value plus playCnt > 31. Sets plrPass=True if Player enters “P”. If plrPass=False—assigns -1 to card ID in hand, calculates playCnt, and sets lastPly = ‘C’. Displays playCnt. Displays card using dspHand(). |
| short vldPlay(const vector<short> &hand, const short crdVal[],const short playCnt) | Parameters: vector<short> hand is a hand containing values -1 through 51; const crdVal[] is an array of card values from plyGame(); unsigned short &playCnt has value 0-31 from plyGame()  Determines if there is a playable card in a particular hand given the current playCnt.  Output: The card ID of the first playable card in the hand, else -1 if no playable cards are found |
| bool hasCrds(vector<short>) | Parameters: Uses a short int vector  Checks each value in the vector to see if any one value is >=0 meaning there is a valid card left to play  Output: returns true if there are one or more cards left, else returns false |
| void thePlay((vector<short> &c\_handC,vector<short> &c\_handP,const short crdVal[], const short crdID[],  const vector<string> &crdName, unsigned short &playCnt,  bool &cmpPass, bool &plrPass, int &pointsC, int &pointsP, char &dealer, char &lastPly, char option) | Parameters: vector<short> &c\_handP is a copy of the Player’s hand from plyGame();vector<short> &c\_handC is a copy of the computer’s hand from plyGame(); const short crdID[] is an array of card id’s for cardID() from plyGame(); const crdVal[] is an array of card values from plyGame() for cardVal();const vector<string> crdName is a vector of card names for dspHand() from plyGame(); unsigned short &playCnt has value 0-31 from plyGame(); bool &plrPass initialized to false from plyGame();int &pointsC is computer’s points total from plyGame(); int &pointsP is player’s points total from plyGame(); char &lastPly holds ‘C’ or ‘P’ from plyGame(); option controls which variable controls the execution order  thePlay() executes plyCrdC() and plyCrdP() using looping constructs. The order of execution is based on either char dealer or char lastPly using char option to determine which variable will be used. Checks if either the player or computer has “passed” and keeps track of the playCnt. Adds “GO” scoring functionality based on lastPly. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Major Variables and Objects** | | | |
| **Function Location** | **Type** | **Name** | **Description** |
| plyGame() | short array | crdID | An array to hold all of the card id’s 0-51; card ID == sub number |
| plyGame() | string vector | crdName | An array to hold all of the card names |
| plyGame() | short array | crdSuit | An array to hold the suit for each card |
| plyGame() | short array | crdVal | An array to hold the numerical value for each card |
| plyGame() | ifstream | names | Input stream object to stream card names from file |
| plyGame() | ifstream | suits | Input stream object to stream card suits from file |
| plyGame() | ifstream | vals | Input stream object to stream numerical card values from file |
| plyGame() | ifstream | ids | Input stream object to stream card ID’s from file |
| plyGame() | character | dealer | ‘C’ for computer, ‘P’ for Player. For determining who is dealer |
| plyGame() | short array | deck | Copy of crdID[] used to hold available card values for the “deck” |
| plyGame() | short vector | handC | Used to hold card values for the computer’s hand; MAX size == 6 |
| plyGame() | short vector | handP | Used to hold card values for the Player’s hand; MAX size == 6 |
| plyGame() | short vector | c\_handC | Copy of handC used to hold card values for the computer’s hand |
| plyGame() | short vector | c\_handP | Copy of handP used to hold card values for the Player’s hand |
| plyGame() | short vector | crib | Used to hold card values for the crib; MAX size == 4 |
| plyGame() | const short | DK\_SIZE | Holds the size of the Deck arrays == 52 |
| plyGame() | integer | pointsC | Holds the computer’s point total; |
| plyGame() | integer | pointsP | Holds the Player’s point total; |
| plyGame() | unsigned short | stCrdID | Holds the starting card’s ID number |
| plyGame() | bool | isStJak | Holds whether the starting card is a Jack. True if a Jack. |
| plyGame() | const float | NIB\_MUL | The nibs multiplier == 1.2; Used as a bonus |
| plyGame() | unsigned short | playCnt | Holds the count for The Play; values 0-31 |
| plyGame() | bool | cmpPass | Initialized to false; indicates if computer has “passed” during play |
| plyGame() | bool | plrPass | Initialized to false; indicates if Player has “passed” during play |
| plyGame() | character | lastPly | ‘C’ for computer, ‘P’ for Player. For determining who played last |