ECE 175: Computer Programming for Engineering Applications
Final Project: *Two pairs or Better*
Due Date: Monday December 6, 2021 at 4.00 pm, via D2L dropbox

All due date: Group sign up - Tuesday November 23, 2021 by 11.59 pm
Mid-project - Tuesday November 23, 2021 at 11.59 pm
Final project - Monday December 6, 2021 at 4.00 pm

Administrative details:

1.1 Group: You can work on this final project **by yourself, or** you can **work in a group of (maximum) 2 students.** If you choose to work as a group, use collaborative tools to facilitate the project development, such as replit.com, google doc, etc.

Sign up your team for the final project here: https://forms.gle/UyhcAqDBJZuvTnNo9

The team/group must be signed-up by 11.59 pm on November 23, 2021. No group can be formed after this date (unless approved by an instructor). Sign up even if you will work by yourself.
You can check that your group signed up properly at
https://docs.google.com/spreadsheets/d/1rQzXrz1E80tkL1S4US2d7z9t6AGU3D-P-owpYmNjnBE/edit?usp=sharing

1.2 Intermediate Code and Design Submission

To keep track of your progress, an intermediate code submission is required. The deadline for the intermediate code submission on D2L is *Tuesday November 23, 2021 at 11.59 pm*.
Late submission policy: 20% deduction per day.
Submit:

1. A high-level design of your code. You are free to use any format that makes sense to your team (e.g., a flowchart with all functions and how the game proceeds, pseudo-code, etc.).
2. An initial code implementation, with
   - a deck of cards being implemented as a linked list
   - a shuffling function that shuffles the deck.
   - a player hand implemented as a linked list
   - a print function that can be used to print the deck or the 5-card player's hand.
3. A skeleton of the remaining code containing all the function prototypes and comments on the input/output of the functions and their intent.

1.3 Final Code Submission: The project is due on *Monday December 6, 2021 at 4.00 pm*. Both team members must submit the code in the designated Dropbox on D2L.
Late submission policy: 20% deduction per day.
The **last day** of final project submission is *by 4.00 PM on Wednesday December 8, 2021*. After this date/time, the final project will NOT be accepted/graded.

1.4 Academic Integrity Policy: Each team is expected to submit its own code. You may ask others for advice and/or discuss the project in general. However, you/your group MUST

WRITE YOUR OWN CODE.

If *any part* of the code submitted by different students or groups of students is identical, ALL involved parties will receive zero credit on the entire project and one letter reduction in their final grade. **This policy will be very aggressively enforced.** ALL submitted code will be checked with a plagiarism detection tool (http://theory.stanford.edu/~aiken/moss/).

1.5 Suggestions

a) Spend time designing your code and use modular programming. Create all function prototypes and describe what they do before developing your code. Create pseudocode of your program.
b) Determine test cases for your functions and your overall code to ensure proper functionality.
c) Write well-documented code.

Project description – Two pairs or Better

Two pairs or Better is a modified poker game in which the goal is to have a 5-card hand that wins the highest payout. The player first places the bet before being dealt a 5-card hand. The player then decides the cards in his/her hand to keep/discard. The discarded cards are then replaced by new cards drawn from the deck. The 5-card hand is then used to decide whether the player wins the payout.

For a deck of 52 cards, 13 faces/ranks are 1(Ace), 2 – 10, Jack (J), Queen (Q), and King (K). Each face has 4 suits: **c**lubs (♣), **s**pades (♠), **h**earts (♥), **d**iamonds (♦), the followings are the 5-card poker hands from highest to lowest ranking:

1. **Royal Flush**: the 5-card hand has face values of 10, J, Q, K, A and all cards have the same suit such as 10♠ J♠ Q♠ K♠ A♠
2. **Straight Flush**: the 5-card hand has face values that are in a sequence and all cards have the same suit such as J♥ 10♥ 8♥ 9♥ 7♥
3. **Four of a Kind**: the 5-card hand has 4 cards of the same face/rank value such as 10♣ 2♦ 10♥ 10♦ 10♠
4. **Full House**: the hand has 3 cards of the same face value and 2 cards of another face value such as 2♣ 2♦ 10♥ 10♦ 10♠
5. **Flush**: all 5 cards have the same suit with any face value (face values are not in a sequence) such as 5♣ 10♣ 4♣ J♣ 8♣
6. **Straight**: the 5-card hand has face values that are in a sequence but not of the same suit such as A♦ 2♦ 3♠ 4♥ 5♣
7. **Three of a Kind**: the 5-card hand has 3 cards of the same face value such as 2♣ 6♦ 10♥ 10♦ 10♠
8. **Two Pairs**: the 5-card hand has 2 pairs (a pair of one face and the second pair of another face) such as 2♣ 2♦ 10♠ 10♦ 5♦

Payout:

The player wins the bet if he/she gets at least two pairs or better with the following payout:

If a user places ***X*** coins for the bet,	***Pay***
Royal Flush	250****X***
Straight Flush	100****X***
Four of a Kind	50****X***
Full House	25****X***
Flush	10****X***
Straight	5****X***
Three of a Kind	4****X***
Two Pair	2****X***

Project requirements: Write an interactive C program for the card game – Two pairs or Better using linked lists for the card deck, as well as the player's hand.

Not using linked lists in your final project code results in 0 point for this final project.

2.1) Deck of 52 cards and player and dealer hands

2.1a) To represent cards, use the following struct type:

```
typedef struct card_s {
        char suit;
        int face;
        struct card_s *next;
} card;
```

Each card consists of a suit [**c**lubs (♣), **s**pades (♠), **h**earts (♥), or **d**iamonds (♦))
a face (1(Ace), 2 – 10, Jacks (J), Queens (Q), and Kings (K))

Note: you can modify the above struct (i.e., modify the data type of member(s), add more members, make variable an array, etc.) but you cannot remove the already existing members.

2.1b) The 52-card deck must be implemented using a dynamic list (linked list) of cards. The cards drawn from the deck MUST be removed from the list.

2.1c) Your program then shuffles the deck. One algorithm that you may use to shuffle the deck.

(a) For each card in the deck, get a random number in the range of 0 to 51 to be used as the index of the element to swap with that card , i.e.

if the first node holds the Jack of clubs (J♣) and the random number generated was 24, and the 24th node holds the 9 of diamonds (9♦), then after the first swap, the first node now holds the 9 of diamonds (9♦), and the 24th node now holds the Jack of clubs (J♣). Then proceed to the second node, find a random index of a card to swap with, and swap those cards, etc.

(b) Repeat step (a) at least 10,000 times.

Once your code is working, seed the random number generator with a call to time() with srand(). [see sec 2.24 Random numbers in your zyBooks]

2.1d) The player's hand is implemented using the dynamic list (linked list) of cards. The list is initially populated with the cards drawn from the deck.

Note: Card(s) added to the player's hand (drawn from the deck) must be added to that player's linked list correctly and MUST be removed from the deck.

2.1e) The free() function should be used to free cards that have been played/discarded.2.1f) The deck should be recreated and shuffled if it has fewer than 20 cards.

2.2 Bet

2.2a) The player starts the game with 100 coins. The game terminates if a) the player loses all coins or b) the player enters -1 when your game/code asks for a bet.

2.2b) The minimum bet is 1 coin, and the maximum bet is number of coins that the player has.

2.3 Game

2.3a) The player places a bet (see 2.2 on page 3 for details on the bet).

2.3b) The player is dealt a 5-card hand.

2.3c) The player chooses the cards that he/she wants to keep/discard. The goal is to have the hand that can win the highest payout (see the winning ranking and payout on page 2).

2.3d) The discarded cards will then be replaced by cards drawn from the deck.

Note: the cards should be properly removed from the deck (linked list) and added to the player's hand (linked list) properly.

2.3e) From the player's hand, the winning condition is decided, and the bet is handled accordingly if the player win or lose. See page 2 for the payout.

2.3f) Each round ends with the player wins or loses a bet. The next round continues as long as the players have enough coins to place a bet, or they enter -1 when your program asks for a bet.

Reminder: the deck should be recreated if it has fewer than 20 cards.

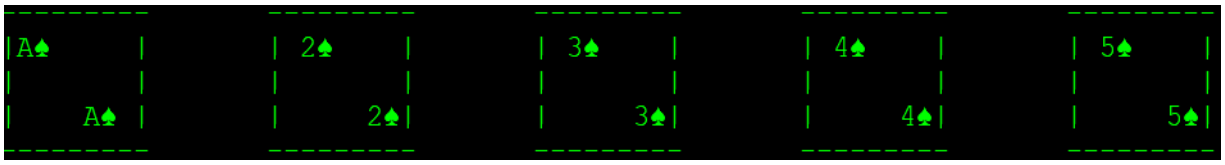
See sample code executions starting on page 7

Optional Features for extra credit:

Note: Each optional feature must be correctly and fully functional in your code to get the extra credit for that part. **No partial credit will be given for somewhat functioning feature(s).**

Make sure that your program is working correctly before attempting to do these bonus features.

1) Graphics (4 pts): Add graphics to your game. Using ASCII art, your code can display the cards as shown below. Note: this is <u>minimum</u> requirement to get these bonus points.
   Note: if your code does not print all 5 cards on the same line as shown below, you will not get bonus points for this part.



2) Suggestions for the player (5 pts)
   Your program suggests, to a player, the cards to keep to win the most payout. For example, if the hand contains
   - one pair (two cards with same faces), your code should suggest the player to keep those 2 cards.
   - Two pairs or Four of a kind, your code should suggest the player to keep those 4 cards.
   - Three of a kind, your code should suggest the player to keep those 3 cards.
   - Straight, Flush, Full house, Straight or Royal Flush, your code should suggest the player to keep those 5 cards.

3) Game statistics (3 pts)
   Keep statistics on the number of rounds played, number of rounds that the player won and lost, the number of rounds with each type of winning. The statistics should be written and appended into the text file at the end of each game (when the player quits or has no coin left).

ECE175--Fall 2021 (Two pairs or Better)
Grading Rubric for Final Project

As stated in the final project handout, linked list must be used for this project. 0 point for final project if linked list is not used.

	Criterion	Maximum Points	Exemplary (100%)	Proficient (75%)	Marginal (20%)	Unacceptable (0%)
	Requirements		**Mid-project (Intermediate)**			
1	Create Deck	5	The deck of 52 cards is created correctly using the linked list. The list is properly initialized and can be traversed. All cards are present in the deck	Deck is implemented as a linked list. However, the implementation is incomplete, not all cards are present or the list is not properly traversed	Some skeleton if a list exists but the code is not properly compiled	The deck is not created using the list or no deck created
2	Deck or hand printing	3	The function can correct print the deck or the 5-card hand	Deck or 5-card hand is printed but the card format does not appear properly.	There was an attempt to do this but not correctly function	No implementation
3	Deck Shuffling	3	The shuffling function produces a new sequence of cards correctly	Deck is partially shuffled but it does not appear sufficiently random	There was an attempt to do this but not correctly function (list is broken after shuffling, …)	The shuffling is not done corrrectly or No Shuffling
4	Card Dealing	5	The 5-card plyer's hand is created correctly. The linked list is used for the player's hand. The cards are removed from the linked list of the deck (free () should be used) OR the cards are moved from the linked list of the deck to the linked list of the player's hand (moving the nodes)	The 5-card plyer's hand is created correctly but no proper code to handle node removal and add or node movings from the linked list of the deck to the linked list of the player's hand	There was an attempt to do this but not correctly function	No implementation
5	Pseudo-code and code skeleton	4	A pseudo-code is presented that accounts for all the game mechanics. Code is sufficiently modular and function prototypes are prototyped.	A pseudo-code is presented,but several game functionalities are not accounted for. The code is not modular and only a few functions are prototyped.	The pseudo-code is scarce and lacks essential details. Hardly any functions are prototyped.	No pseudo-code or functions are defined.
	Game		**Exemplary (100%)**	**Proficient (75%)**	**Marginal (20%)**	**Unacceptable (0%)**
6	Let a user enter a name and propely print the info of a game (see sample code execition)	2	Info and player's name is corectly displayed	Partially complete (i.e., no name displayed, info displayed is incomplete)	There was an attempt to do this but not correctly function	No implementation
7	Deck recreated and shuffled when number of cards of the deck are fewer than 20	3			There was an attempt to do this but not correctly function	No implementation
8	Bet	2	The program only allows a user to enter between 1 and maximum coins that the player has and will continue asking until the user enters a valid numbers for a bet	Mainly working but not 100% correct (i.e., missing checking number < 1 or missing checking number > maximum coins)	There was an attempt to do this but not correctly function	No implementation
	Bet (-1 to end the game)	3	The program ends corectly when the player enters -1 when the program asks for a bet		There was an attempt to do this but not correctly function (the program does not end)	No implementation
9	Print the player's hand at the beginning of the round	2	The player's hand is printed correctly		There was an attempt to do this but not correctly function	No implementation
10	Player turn: cards held - only accepting card number 1-5	2	The program allows a user to enter only 1-5. If a user does not, continue asking until the user enters a valid card number	Mainly working but not 100% correct (i.e., missing checking number < 1 (except -1) or missing checking number > 5)	There was an attempt to do this but not correctly function	No implementation
11	Player turn: cards held - accept cards held until -1 is entered or all 5 cards are held	3	continue asking a card number to be held until -1 is entered or all 5 cards are held.		There was an attempt to do this but not correctly function	No implementation
12	Replacement of discraded card(s)	5	Discarded cards are replaced by cards drawn from the deck correctly. The linked list of the deck and the player's hand are handled correctly	Discarded cards are replaced by cards drawn from the deck correctly. The linked list of the deck and the player's hand are NOT handled correctly	There was an attempt to do this but not correctly function	No implementation

13	Print the player's hand after the player keep/discard card(s)	2	The player's hand is printed correctly		There was an attempt to do this but not correctly function	No implementation
14	winning ranking	23	if the program can correclty decide each winning ranking when a hand contains (3 pts) Royal Flush, (3 pts) Staright Flush, (3 pts), Four of a Kind, (3 pts) Full House, (2 pts) Flush, (3 pts) Straight, (3 pts) Three of a Kind, (3 pts) Two Pairs			
15	winning or losing bet	6	if the program can decide whether the player win the bet and properly add or deduct the number of coins based on the winning ranking	if the program can decide whether the player win the bet BUT wrongly add or deduct the number of coins	There was an attempt to do this but not correctly function	No implementation
16	next round	3	the next round continues properly if a player still has coins		There was an attempt to do this but not correctly function (only one round can be played even though the player still has coins)	No implementation
17	new game	3	if the player has 0 coins, the game ends correclty. Linked lists are properly free	there is a major attempt on this part and it is 75% working or Linked lists are not free	There was an attempt to do this but not correctly function or not working	No implementation
18	Game Interface	6	The interface is intuitive. The user is able to play the game using the nagivation that the program provides	The interface is intuitive for the most part. Some difficulty in understanding the game navigation.	The interface is counter-intuitive. Navigation options are not clearly stated. Interface limitations prevent proper game functionality.	No implementation
	Program Design					
19	Code modularity	5	The code is logically divided to several functions that implement *important* functionality	The code is modular but further simplification could have been attempted		Code is not modular (all statements are written in main)
20	Code documentation	3	The code is properly documented. The input/output and goal of every function is adequately described. Comments are provided for various parts of the code	The code is partially documented	The code is scarcely documented	No documentation is provided
21	Compilation	7	Code succesfully compiles without errors or warnings. The code does not hang while in execution	The code succesfully compiles, but some conditions may make it hang (-4 pts)		Code does not compile
	Total	**100**				
	Extra Credit		**Exemplary (100%) - NO PARTIAL CREDIT - only get these extra credits if it works correctly**			
	Graphics	4	See Bonus points section of the final project handout for details			
	Suggestion to player	5	See Bonus points section of the final project handout for details			
	Game statistics	3	See Bonus points section of the final project handout for details			
	Total	**12**				

Note:

- All sample code execution given below are a minimum requirement to give you an idea how the game should be implemented.
- You are encouraged to make your game look more appealing than what is given below. However, your program should display very similar information for a user to be able to play the game.
- It is VERY LIKELY that you will get DIFFERENT code execution (cards will be different) when running your program since rand() is used.

Sample Code Execution 1: demonstrates a) keeps asking a player if he/she enters invalid inputs, b) a few winning conditions: Straight Flush and Flush and c) when a player quits.

Sample Code Execution 2: demonstrates a few winning conditions: Four of a Kind and Full House.

Sample Code Execution 3: demonstrates when a player loses all coins.

Sample Code Execution 4: demonstrates a regular game.

Sample Code Execution 1: **Bold entered by a user**

Note: I do not shuffle the deck for this execution so that I can test a few winning conditions.

Enter your name: **Wilbur**

$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
Wilbur,
Let's play Two pairs or Better
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ Rank of winning $$$$$$$$$

		Pay
Royal Flush	10♠ J♠ Q♠ K♠ A♠	250*bet
Straight Flush	2♣ 3♣ 4♣ 5♣ 6♣	100*bet
Four of a Kind	9♠ 9♣ 9♦ 9♥ ■	50*bet
Full House	9♦ 9♥ 9♠ 3♣ 3♥	25*bet
Flush	■♣ ■♣ ■♣ ■♣ ■♣	10*bet
Straight	4■ 5■ 6■ 7■ 8■	5*bet
Three of a Kind	9♠ 9♣ 9♦ ■ ■	4*bet
Two Pair	K♠ K♦ 6♦ 6♥ ■	2*bet

Note: I use ■ to represent that a card can be any card for that winning rank.

Flush, ■ is used to represent that face of cards can be any value given that they are not in sequential order.

Straight, ■ is used to represent that suit of cards can be any suit

Your program can use other symbols!

********* ********* ********* ********* *********
********* Wilbur, you have 100 coins *********
********* ********* ********* ********* *********
Place your bet (1-100) coins (-1 to quit playing): **0**
Place your bet (1-100) coins (-1 to quit playing): **120**
Place your bet (1-100) coins (-1 to quit playing): **10**

Note: your program should continuously ask if a user does not enter appropriate bet.

You bet 10 coins

Wilbur's hand:
A♠ 2♠ 3♠ 4♠ 5♠

Pick cards (between 1-5) to hold (-1 to stop): **0**
Pick cards (between 1-5) to hold (-1 to stop): **10**
Pick cards (between 1-5) to hold (-1 to stop): **6**

Note: your program should continuously ask if a user enter a value that is NOT -1 or a value that is NOT between 1 and 5.

Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **4**
Pick cards (between 1-5) to hold (-1 to stop): **5**

Note: all cards do not change since a player holds all 5 cards.

Wilbur's hand: (Straight Flush)
A♠ 2♠ 3♠ 4♠ 5♠

since ***Straight Flush*** pays 100*bet and Wilbur bet 10 coins, Wilbur therefore won 1000 coins

You WON 1000 coins and you now have 1100 coins

Hit Enter key to continue: ***Enter key***

Hit Enter key to continue is a code segment that I implement. Your group can do as you see fit to make your game user friendly or easy to follow

********* ********* ********* ********* *********
********* Wilbur, you have 1100 coins *********
********* ********* ********* ********* *********

Place your bet (1-1100) coins (-1 to quit playing): 10

You bet 10 coins

Wilbur's hand:
6♠ 7♠ 8♠ 9♠ 10♠
Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **4**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Note: card #1-4 do not change since a user hold those cards. The 5th card is new (drawn from the deck)

Wilbur's hand: (Flush)
6♠ 7♠ 8♠ 9♠ J♠

since ***Flush*** pays 10*bet and Wilbur bet 10 coins, Wilbur therefore won 100 more coins

You WON 100 coins and you now have 1200 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Wilbur, you have 1200 coins *********
********* ********* ********* ********* *********

Place your bet (1-1200) coins (-1 to quit playing): **-1**
Goodbye Wilbur

Sample Code Execution 2: **Bold entered by a user**

Note: I create the deck using another way and still do not shuffle the deck for this execution so that I can test a few more winning conditions.

Enter your name: **Jack Black**

$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
Jack Black,
Let's play Two pairs or Better
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ Rank of winning $$$$$$$$$

		Pay
Royal Flush	10♠ J♠ Q♠ K♠ A♠	250*bet
Straight Flush	2♣ 3♣ 4♣ 5♣ 6♣	100*bet
Four of a Kind	9♠ 9♣ 9♦ 9♥ ■	50*bet
Full House	9♦ 9♥ 9♠ 3♣ 3♥	25*bet
Flush	■♣ ■♣ ■♣ ■♣ ■♣	10*bet
Straight	4■ 5■ 6■ 7■ 8■	5*bet
Three of a Kind	9♠ 9♣ 9♦ ■ ■	4*bet
Two Pair	K♠ K♦ 6♦ 6♥ ■	2*bet

********* ********* ********* ********* *********
********* Jack Black, you have 100 coins *********
********* ********* ********* ********* *********

Place your bet (1-100) coins (-1 to quit playing): **1000**
Place your bet (1-100) coins (-1 to quit playing): **10**

You bet 10 coins

Jack Black's hand:
A♠ A♦ A♥ A♣ 2♠
Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **4**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jack Black's hand: (Four of a Kind)
A♠ A♦ A♥ A♣ 2♦

since ***Four of a Kind*** pays 50*bet and Jack Black bet 10 coins, Jack therefore won 500 coins

You WON 500 coins and you now have 600 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jack Black, you have 600 coins *********
********* ********* ********* ********* *********

Place your bet (1-600) coins (-1 to quit playing): **10**

You bet 10 coins

Jack Black's hand:
2♥ 2♣ 3♠ 3♦ 3♥

Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **4**
Pick cards (between 1-5) to hold (-1 to stop): **5**

Jack Black's hand: (Full House)
2♥ 2♣ 3♠ 3♦ 3♥

since ***Full House*** pays 25*bet and Jack Black bet 10 coins, Jack therefore won 250 more coins

You WON 250 coins and you now have 850 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jack Black, you have 850 coins *********
********* ********* ********* ********* *********

Place your bet (1-850) coins (-1 to quit playing): **-1**
Goodbye Jack Black

Sample Code Execution 3: **Bold entered by a user**

Enter your name: **Jane**

$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
Jane,
Let's play Two pairs or Better
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ Rank of winning $$$$$$$$$

		Pay
Royal Flush	10♠ J♠ Q♠ K♠ A♠	250*bet
Straight Flush	2♣ 3♣ 4♣ 5♣ 6♣	100*bet
Four of a Kind	9♠ 9♣ 9♦ 9♥ ■	50*bet
Full House	9♦ 9♥ 9♠ 3♣ 3♥	25*bet
Flush	■♣ ■♣ ■♣ ■♣ ■♣	10*bet
Straight	4■ 5■ 6■ 7■ 8■	5*bet
Three of a Kind	9♠ 9♣ 9♦ ■ ■	4*bet
Two Pair	K♠ K♦ 6♦ 6♥ ■	2*bet

********* ********* ********* ********* *********
********* Jane, you have 100 coins *********
********* ********* ********* ********* *********

Place your bet (1-100) coins (-1 to quit playing): **25**

You bet 25 coins

Jane's hand:
6♥ 9♣ Q♥ 8♦ 7♣
Pick cards (between 1-5) to hold (-1 to stop): -1

Note: all 5 cards are changed since the player does not keep any.

Jane's hand: 8♥ K♣ J♦ 3♣ 3♠

You LOST and you now have 75 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 75 coins *********
********* ********* ********* ********* *********

Place your bet (1-75) coins (-1 to quit playing): 75

You bet 75 coins

Jane's hand:
4♥ 7♠ 10♦ Q♦ 5♦
Pick cards (between 1-5) to hold (-1 to stop): **4**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jane's hand: 8♠ 10♥ 2♠ Q♦ 4♦

You LOST and you now have 0 coins

You lost all your coins. Game Over!
Goodbye Jane

Sample Code Execution 4: **Bold entered by a user**

Enter your name: **Jane**

$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
Jane,
Let's play Two pairs or Better
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$ $$$$$$$$$
$$$$$$$$$ Rank of winning $$$$$$$$$

		Pay
Royal Flush	10♠ J♠ Q♠ K♠ A♠	250*bet
Straight Flush	2♣ 3♣ 4♣ 5♣ 6♣	100*bet
Four of a Kind	9♠ 9♣ 9♦ 9♥ ■	50*bet
Full House	9♦ 9♥ 9♠ 3♣ 3♥	25*bet
Flush	■♣ ■♣ ■♣ ■♣ ■♣	10*bet
Straight	4■ 5■ 6■ 7■ 8■	5*bet
Three of a Kind	9♠ 9♣ 9♦ ■ ■	4*bet
Two Pair	K♠ K♦ 6♦ 6♥ ■	2*bet

********* ********* ********* ********* *********
********* Jane, you have 100 coins *********
********* ********* ********* ********* *********
Place your bet (1-100) coins (-1 to quit playing): **25**

You bet 25 coins

Jane's hand:
A♠ Q♥ Q♦ 4♥ 5♣
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jane's hand: 2♥ Q♥ Q♦ 5♦ 8♦

You LOST and you now have 75 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 75 coins *********
********* ********* ********* ********* *********

Place your bet (1-75) coins (-1 to quit playing): **10**

You bet 10 coins

Jane's hand:
3♣ 8♣ 8♠ 10♦ 7♥
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jane's hand: (Two Pair)
K♣ 8♣ 8♠ K♠ 3♥

You WON 20 coins and you now have 95 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 95 coins *********
********* ********* ********* ********* *********

Place your bet (1-95) coins (-1 to quit playing): **15**

You bet 15 coins

Jane's hand:
7♠ 7♣ J♦ 4♠ 9♠
Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **2**

Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jane's hand: 7♠ 7♣ J♦ K♦ 10♥

You LOST and you now have 80 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 80 coins *********
********* ********* ********* ********* *********

Place your bet (1-80) coins (-1 to quit playing): **45**

You bet 45 coins

Jane's hand:
6♠ 10♠ 6♦ 8♥ 3♦
Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jane's hand: (Three of a Kind)
6♠ J♠ 6♦ 2♠ 6♥

You WON 180 coins and you now have 260 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 260 coins *********
********* ********* ********* ********* *********

Place your bet (1-260) coins (-1 to quit playing): 60

You bet 60 coins

Jane's hand:
6♣ 2♣ 3♠ 5♠ J♥
Pick cards (between 1-5) to hold (-1 to stop): **5**
Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jane's hand: 6♣ 2♣ 3♠ 5♥ J♥

You LOST and you now have 200 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 200 coins *********
********* ********* ********* ********* *********
Place your bet (1-200) coins (-1 to quit playing): **100**

You bet 100 coins

Jane's hand:
4♥ K♠ 4♦ A♠ J♠
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **4**
Pick cards (between 1-5) to hold (-1 to stop): **5**
Pick cards (between 1-5) to hold (-1 to stop): **-1**

Jane's hand: A♦ K♠ 7♦ A♠ J♠

You LOST and you now have 100 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 100 coins *********
********* ********* ********* ********* *********

Place your bet (1-100) coins (-1 to quit playing): **1**

You bet 1 coins

Jane's hand:
A♠ 3♠ 7♠ Q♠ 5♠
Pick cards (between 1-5) to hold (-1 to stop): **1**
Pick cards (between 1-5) to hold (-1 to stop): **2**
Pick cards (between 1-5) to hold (-1 to stop): **3**
Pick cards (between 1-5) to hold (-1 to stop): **4**
Pick cards (between 1-5) to hold (-1 to stop): **5**

Jane's hand: (Flush)
A♠ 3♠ 7♠ Q♠ 5♠

You WON 10 coins and you now have 110 coins

Hit Enter key to continue: ***Enter key***

********* ********* ********* ********* *********
********* Jane, you have 110 coins *********
********* ********* ********* ********* *********

The game continues the player loses all coins or decides to quit!