



## Intro to JavaScript Week 6 Coding Assignment

Points possible: 100

URL to GitHub Repository:

[https://github.com/WhitneyHendrickson94/Week6CodingAssignment\\_WAR.git](https://github.com/WhitneyHendrickson94/Week6CodingAssignment_WAR.git)

URL to Your Coding Assignment Video:

```
10 // Create an array to hold the 52 cards
11 let deck = [];
12
13 // Create an array to hold the 4 suits
14 let suits = ['Hearts', 'Diamonds', 'Clubs', 'Spades'];
15
16 // Create an array to hold the 13 ranks
17 let ranks = ['Ace', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'Jack', 'Queen', 'King'];
18
19 // Create the 52 cards
20 for (let i = 0; i < suits.length; i++) {
21   for (let j = 0; j < ranks.length; j++) {
22     let card = suits[i] + ranks[j];
23     deck.push(card);
24   }
25 }
26
27 // Shuffle the deck
28 function shuffleDeck(deck) {
29   for (let i = deck.length - 1; i > 0; i--) {
30     let j = Math.floor(Math.random() * (i + 1));
31     [deck[i], deck[j]] = [deck[j], deck[i]];
32   }
33 }
34
35 // Deal the cards
36 function dealCards(deck, num) {
37   let cards = [];
38   for (let i = 0; i < num; i++) {
39     cards.push(deck.pop());
40   }
41   return cards;
42 }
43
44 // Play a card
45 function playCard(player, card) {
46   // Check if the card is in the player's hand
47   let index = player.hand.findIndex(c => c === card);
48   if (index > -1) {
49     player.hand.splice(index, 1);
50     return true;
51   }
52   return false;
53 }
54
55 // Determine the winner
56 function determineWinner(player1, player2) {
57   // Get the player's cards
58   let cards1 = player1.hand;
59   let cards2 = player2.hand;
60
61   // Calculate the player's score
62   let score1 = 0;
63   let score2 = 0;
64
65   for (let i = 0; i < cards1.length; i++) {
66     let card = cards1[i];
67     let rank = card.split('')[0];
68     let value = rank === 'Ace' ? 1 : rank === 'Jack' ? 10 : rank === 'Queen' ? 10 : rank === 'King' ? 10 : rank;
69     score1 += value;
70   }
71
72   for (let i = 0; i < cards2.length; i++) {
73     let card = cards2[i];
74     let rank = card.split('')[0];
75     let value = rank === 'Ace' ? 1 : rank === 'Jack' ? 10 : rank === 'Queen' ? 10 : rank === 'King' ? 10 : rank;
76     score2 += value;
77   }
78
79   // Determine the winner
80   if (score1 > score2) {
81     return 'Player 1 wins!';
82   } else if (score2 > score1) {
83     return 'Player 2 wins!';
84   } else {
85     return 'Tie!';
86   }
87 }
88
89 // Main function
90 function main() {
91   // Shuffle the deck
92   shuffleDeck(deck);
93
94   // Deal the cards
95   let cards1 = dealCards(deck, 5);
96   let cards2 = dealCards(deck, 5);
97
98   // Play the cards
99   let player1 = {
100     hand: cards1,
101     score: 0
102   };
103   let player2 = {
104     hand: cards2,
105     score: 0
106   };
107
108   // Determine the winner
109   let winner = determineWinner(player1, player2);
110   console.log(winner);
111 }
112
113 // Call the main function
114 main();
```



**Instructions:** In Visual Studio Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

## Coding Steps:

For the final project you will be creating an automated version of the classic card game *WAR*. You do not need to accept any user input, when you run your code, the entire game should play out instantly without any user input.

There are many versions of the game *WAR*, but in this version there are only 2 players and you don't need to do anything special when there is a tie on a round.

Think about how you would build this project and write your plan down. Consider classes such as Card, Deck, and Player and what fields and methods they might each have. You can implement the game however you'd like (i.e. printing to the console, using alert, or some other way). The completed project should, when run, do the following:

- Deal 26 Cards to two Players from a Deck.
- Iterate through the turns where each Player plays a Card
- The Player who played the higher card is awarded a point
  - o Ties result in zero points for both Players
- After all cards have been played, display the score and declare the winner.

Write a Unit Test using Mocha and Chai for at least one of the functions you write.

## Screenshots of Code:



# PROMINEO TECH

```
<> Week6.html  JS Week6.js x  JS Week6_Tests.js  <> Week6_Tests.html
JS Week6.js > ...
1  //Set Variables to player scores to reference later in the game development
2
3  let playerOneScore = 0;
4  let playerTwoScore = 0;
5
6  //create the smallest piece of our game which is a card by creating a card class
7  class Card {
8      constructor(suit,value) {
9          this.suit = suit;
10         this.value = value;
11     }
12
13     //needed to get play value to compare value of played cards each round
14     getPlayValue() {
15         switch (this.value){
16             case 'A':
17                 return 1;
18             case Number.value:
19                 return Number;
20             case 'J':
21                 return 11;
22             case 'Q':
23                 return 12;
24             case 'K':
25                 return 13;
26         }
27         return (this.value);
28     }
29 }
30
31 //Deck is the next piece of our game - created a class which was going to be constructed of cards (held in an array)
32 class Deck {
33     constructor() {
34         this.cards = [];
35     }
36
37     //created createDeck method by using for loops to iterate and combine suit and value arrays
38     createDeck() {
39         let suit = ['clubs', 'diamonds', 'hearts', 'spades'];
40         let value = ["A", 2, 3, 4, 5, 6, 7, 8, 9, 10, "J", "Q", "K"];
41
42         for (let i = 0; i < suit.length; i++) {
43             for (let j = 0; j < value.length; j++) {
44                 this.cards.push(new Card(suit[i], value[j]));
45             }
46         }
47     }
48 }
```



# PROMINEO TECH

```
<> Week6.html  JS Week6.js  JS Week6_Tests.js  <> Week6_Tests.html
JS Week6.js > ...
45     }
46     //created shuffleDeck method
47     shuffleDeck() {
48         for(let i = this.cards.length - 1; i > 0; i--){
49             let j = Math.floor(Math.random() * (i + 1));
50             let x = this.cards[i];
51             this.cards[i] = this.cards[j];
52             this.cards[j] = x;
53         }
54     }
55 }
56 }
57
58 //Create Player class: player needs to have name and their own array to hold their own deck of cards
59 class Player {
60     constructor(name) {
61         this.playerName = name;
62         this.playerCards = [];
63     }
64 }
65
66
67
68 //Game class is the overarching piece of our game that holds all of the functionality
69 class Game {
70     constructor() {
71         this.players = [];
72     }
73
74     //Basic Game Set Up: players added, create deck, shuffle, deal
75     setUpPlay(playerOneName, playerTwoName){
76         //Put players in the game
77         this.players.push(new Player(playerOneName));
78         this.players.push(new Player(playerTwoName));
79
80         //Game Set Up : Create new instance of our Deck class and called our create deck and shuffle deck method
81         let deck1 = new Deck();
82         deck1.createDeck();
83         deck1.shuffleDeck();
84         //Dealt cards to each player by splitting the shuffled deck in half
85         this.players[0].playerCards = deck1.cards.slice(0, 26);
86         this.players[1].playerCards = deck1.cards.slice(26, 52);
87     }
}
```



# PROMINEO TECH

```
< > Week6.html JS Week6.js X JS Week6_Tests.js < > Week6_Tests.html
JS Week6.js > ...
88 //Used the playRounds method to create action of player each round
89 playRounds(){
90     //Iterated thorough rounds with a for loop for each player to play their card
91     for (let i = 0; i < 26; i++){
92         this.players[0].playerCards[i];
93         this.players[1].playerCards[i];
94         console.log('Player one's card is: ${this.players[0].playerCards[i].value} of ${this.players[0].playerCards[i].suit} `');
95         console.log('Player two's card is: ${this.players[1].playerCards[i].value} of ${this.players[1].playerCards[i].suit}`');
96         //Had to account for getting actual playValue before comparing their values to award points for each round
97         this.players[0].playerCards[i].getPlayValue();
98         this.players[1].playerCards[i].getPlayValue();
99         //Nested if/If else statements were used to compare played card values and award points to players
100         if(this.players[0].playerCards[i].getPlayValue() > this.players[1].playerCards[i].getPlayValue()){
101             console.log('Player One wins this round!');
102             playerOneScore += 1;
103
104         }else if(this.players[0].playerCards[i].getPlayValue() < this.players[1].playerCards[i].getPlayValue()){
105             console.log('Player Two wins this round!');
106             playerTwoScore += 1;
107
108         }else{
109             console.log('It's a tie this round.');
```



## Screenshots of Running Application:

Player one's card is: 3 of spades	<a href="#">Week6.js:94</a>
Player two's card is: 7 of diamonds	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: J of hearts	<a href="#">Week6.js:94</a>
Player two's card is: Q of diamonds	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: Q of clubs	<a href="#">Week6.js:94</a>
Player two's card is: 3 of clubs	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 9 of hearts	<a href="#">Week6.js:94</a>
Player two's card is: 5 of diamonds	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 7 of hearts	<a href="#">Week6.js:94</a>
Player two's card is: 4 of diamonds	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: Q of spades	<a href="#">Week6.js:94</a>
Player two's card is: 2 of clubs	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 5 of clubs	<a href="#">Week6.js:94</a>
Player two's card is: 5 of hearts	<a href="#">Week6.js:95</a>
It's a tie this round.	<a href="#">Week6.js:109</a>
Player one's card is: 2 of spades	<a href="#">Week6.js:94</a>
Player two's card is: 6 of diamonds	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: K of clubs	<a href="#">Week6.js:94</a>
Player two's card is: 10 of spades	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 3 of hearts	<a href="#">Week6.js:94</a>
Player two's card is: A of hearts	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 2 of diamonds	<a href="#">Week6.js:94</a>



# PROMINEO TECH

Player one's card is: 5 of spades	<a href="#">Week6.js:94</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: A of diamonds	<a href="#">Week6.js:94</a>
Player two's card is: K of spades	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: 8 of hearts	<a href="#">Week6.js:94</a>
Player two's card is: K of diamonds	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: K of hearts	<a href="#">Week6.js:94</a>
Player two's card is: J of diamonds	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 2 of hearts	<a href="#">Week6.js:94</a>
Player two's card is: 4 of clubs	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: J of clubs	<a href="#">Week6.js:94</a>
Player two's card is: 8 of clubs	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 9 of clubs	<a href="#">Week6.js:94</a>
Player two's card is: 10 of hearts	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: 9 of spades	<a href="#">Week6.js:94</a>
Player two's card is: Q of hearts	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: 4 of hearts	<a href="#">Week6.js:94</a>
Player two's card is: A of spades	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: A of clubs	<a href="#">Week6.js:94</a>
Player two's card is: 7 of clubs	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: 6 of spades	<a href="#">Week6.js:94</a>
Player two's card is: 6 of hearts	<a href="#">Week6.js:95</a>
It's a tie this round.	<a href="#">Week6.js:109</a>
Player one's card is: 9 of diamonds	<a href="#">Week6.js:94</a>
Player two's card is: 8 of spades	<a href="#">Week6.js:95</a>
Player One wins this round!	<a href="#">Week6.js:101</a>
Player one's card is: 6 of clubs	<a href="#">Week6.js:94</a>
Player two's card is: 10 of clubs	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: 3 of diamonds	<a href="#">Week6.js:94</a>
Player two's card is: 8 of diamonds	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: 5 of spades	<a href="#">Week6.js:94</a>
Player two's card is: 7 of spades	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player one's card is: 4 of spades	<a href="#">Week6.js:94</a>
Player two's card is: 10 of diamonds	<a href="#">Week6.js:95</a>
Player Two wins this round!	<a href="#">Week6.js:105</a>
Player Two wins the game!	<a href="#">Week6.js:120</a>
The final score is Player One: 10 Player Two: 14	<a href="#">Week6.js:124</a>
>	