



Intro to JavaScript Week 6 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

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*.

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 ran, 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 either Player
 - After all cards have been played, display the score.

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

Screenshots of Code: Screenshots of Running Application:

passes 2 failures duration: 1.24s

```
 1 // codingame.com
 2 // week5/day1
 3 // week5/day1.x
 4 // testfile
 5
 6 // expect = chai.expect
 7
 8 describe("My Function", function() {
 9   it("should return an object", function() {
10     expect("player").to.be.an("object");
11   });
12   it("it should return a player", function() {
13     var x = new Player();
14     expect(x).to.be.instanceOf(Player);
15   });
16 });
17
```

index

My Functions

1 passing 1 failing 0 duration: 1.24s

it should return an object ✖

it should return a player ✖

node-v16.14.0.pkg

Show All

The screenshot shows a browser window with developer tools open, displaying a game's source code and a running game session.

Code:

```
weekSix.js X weekSix_Test.js X tests.html node-v16.10.0.js
```

```
1 // weekSix.js
2
3 class GamePlay {
4   constructor() {
5     this.theCards = null;
6   }
7
8   gamePlay() {
9     let cards = this.theCards;
10    ...
11  }
12}
13
14function generateDeck() {
15  let newDeck = [];
16  let i = Math.floor(Math.random() * theCards.length); // rand index
17  let card = theCards[i];
18  newDeck.push(card);
19  theCards.splice(i, 1);
20  ...
21}
22
23let playerOne = new Player("Player One", theCards);
24let playerTwo = new Player("Player Two", theCards);
25
26playerOne.setCard();
27playerTwo.setCard();
28
29for (let i = 0; i < 10; i++) {
30  console.log(`Player one has ${playerOne.cards[i].face} of ${playerOne.cards[i].suit}`);
31  console.log(`Player two has ${playerTwo.cards[i].face} of ${playerTwo.cards[i].suit}`);
32}
33
34if (playerOne.cards[1].value > playerTwo.cards[1].value) {
35  console.log(`Player one wins!`);
36} else if (playerOne.cards[1].value == playerTwo.cards[1].value) {
37  console.log(`It's a tie! Both zero points!`);
38} else {
39  console.log(`Player two wins!`);
40}
41
42if (playerOne.points > playerTwo.points) { //announce the winner
43  alert(`Player one wins the game!`);
44} else if (playerTwo.points > playerOne.points) {
45  alert(`Player two wins the game!`);
46}
47else {
48  alert(`Player two wins the game!`);
49}
50
51}
52
53gamePlay();
```

Console Output:

```
[1] Console - What's New | Issues
Highlights from the Chrome 93 update

Throttling web socket requests
The Network panel now supports throttling web socket requests.

Recorder panel improvements
Wait for elements to be visible in the viewport before reporting the next step.
```

Network Tab:

```
node-v16.10.0.js
```

Elements Tab:

```
Course: Introduction to JS
```

Resources Tab:

```
Week 6 - Mocha Test
```

Performance Tab:

```
codingSix.html
```

```
 39      }
40    }
41  }
42}
43
44function gamePlay(cards) {
45  let handIndex = new Deck();
46  let playerOne = new Player(cards, handIndex);
47  let playerTwo = new Player(cards, handIndex);
48
49  for (let i = 1; i < cards.length; i++) {
50    console.log(`Player ${i} card ${i}: ${playerOne.cards[i].face} of ${playerOne.cards[i].suit}`);
51    console.log(`Player ${i} card ${i}: ${playerTwo.cards[i].face} of ${playerTwo.cards[i].suit}`);
52  }
53  if ((playerOne.cards[0].value > playerOne.cards[1].value) ||
54    (playerOne.cards[1].value > playerOne.cards[0].value)) {
55    playerTwo.points += 1;
56  }
57}
58
59else if ((playerOne.cards[0].value == playerOne.cards[1].value) ||
60          (playerTwo.cards[0].value == playerTwo.cards[1].value)) {
61  console.log(`It's a tie! Both players have zero points!!!`);
62}
63
64if (playerOne.points > playerTwo.points) { //announce the winner
65  return `Player One wins the game!!!`;
66}
67else {
68  return `Player Two wins the game!!!`;
69}
70}
71
```

The screenshot shows a browser window with a SonarQube report for a file named 'codingSix.js'. The report lists 26 issues, mostly related to card values and suits. The browser's address bar shows the file path: 'Usernameshankar/Documents/Weeks/Week49/Project/codingSix.js'. The tabs at the top of the browser include 'Course: Introduction to ...', 'File | Usernameshankar/Documents/Weeks/Week49/Project/codingSix.js', 'Week 6 - Media Test', and 'codingSix.js'.

```
js week8Card.js > gamePlay
1 class Player {
2   constructor(cards) {
3     this.points = 0;
4     this.cards = cards;
5   }
6 }
7
8 class Card {
9   constructor(suit, value, face) {
10    this.suit = suit;
11    this.value = value;
12    this.face = face;
13  }
14 }
15
16
17 class Deck {
18   constructor(){
19    this.theCards = [];
20
21    let suit = ["hearts", "clubs", "spades", "diamonds"];
22    let value = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14];
23    let face = ["two", "three", "four", "five", "six", "seven", "eight", "nine", "ten", "jack", "queen", "king", "ace"];
24
25    // build the deck
26    for (let i = 0; i < suit.length; i++) {
27      for (let j = 0; j < value.length; j++) {
28        this.theCards.push(new Card(suit[i], value[j], face[j]));
29      }
30    }
31
32    // shuffle
33    this.theCards = this.theCards.sort(() => Math.random() - 0.5);
34  }
35
36
37
38
39
40
41 function startPlay(){
42   let handDeck = new Deck();
43   let m = Math.floor(handDeck.theCards.length / 2); //find index
44   let playOrder = new Player(handDeck.theCards.slice(0,m));
45   let playOrder2 = new Player(handDeck.theCards.slice(m));
46   //this is the dealer
47
48   for (let i = 0; i < m; i+=1){
49     const card = handDeck.theCards[i];
50     const card2 = handDeck.theCards[m+i];
51     playOrder.addCard(card);
52     playOrder2.addCard(card2);
53   }
54 }
```



PROMINEO TECH

URL to GitHub Repository: <https://github.com/hannah1comb/weekSixWar>