PROMINEO TECH

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:

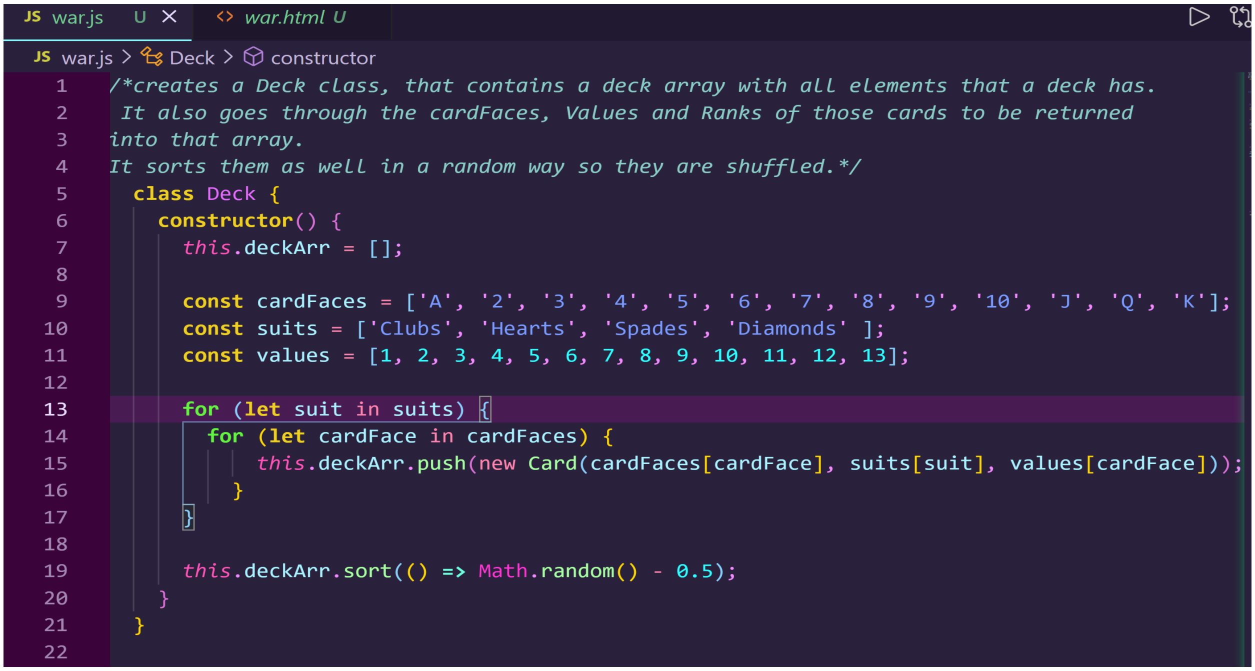
For the final project you will be creating an automated version of the classic card game *WAR.*

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:

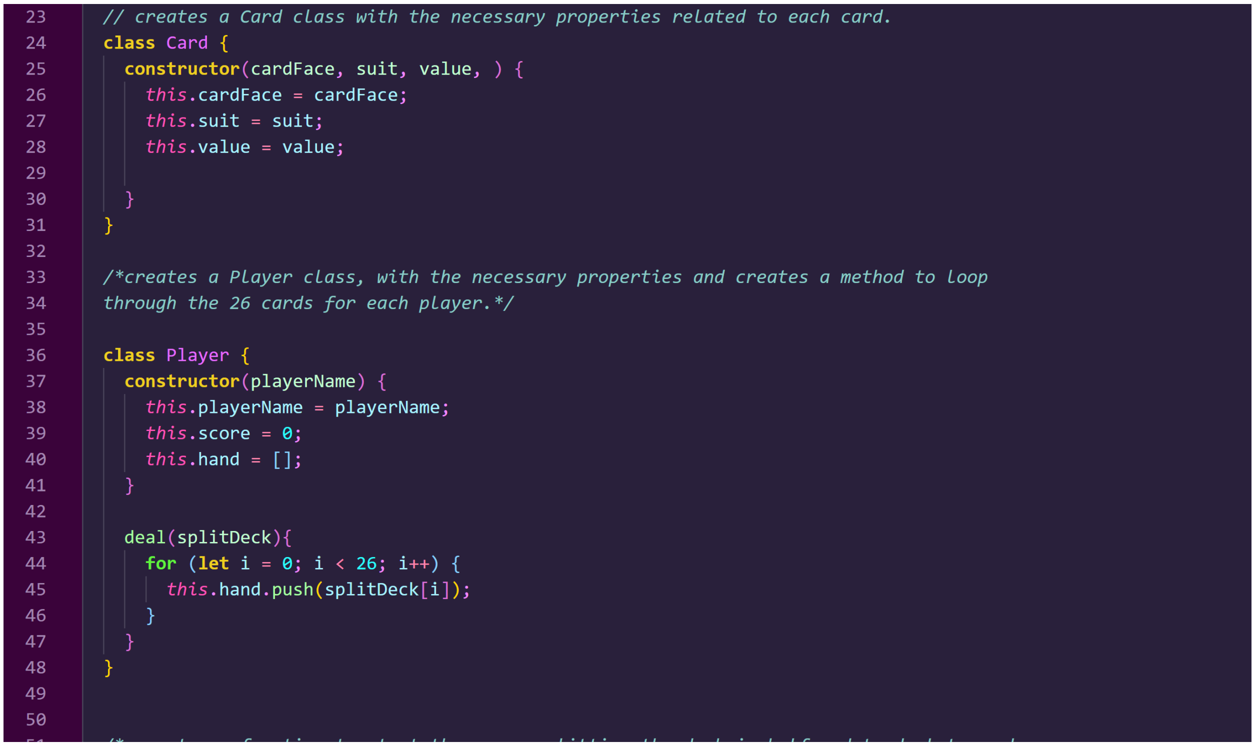
- Deal 26 Cards to two Players from a Deck.
- Iterate through the turns where each Player plays a Card
- The Player who played the higher card is awarded a point
  - Ties result in zero points for either Player
- After all cards have been played, display the score.

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

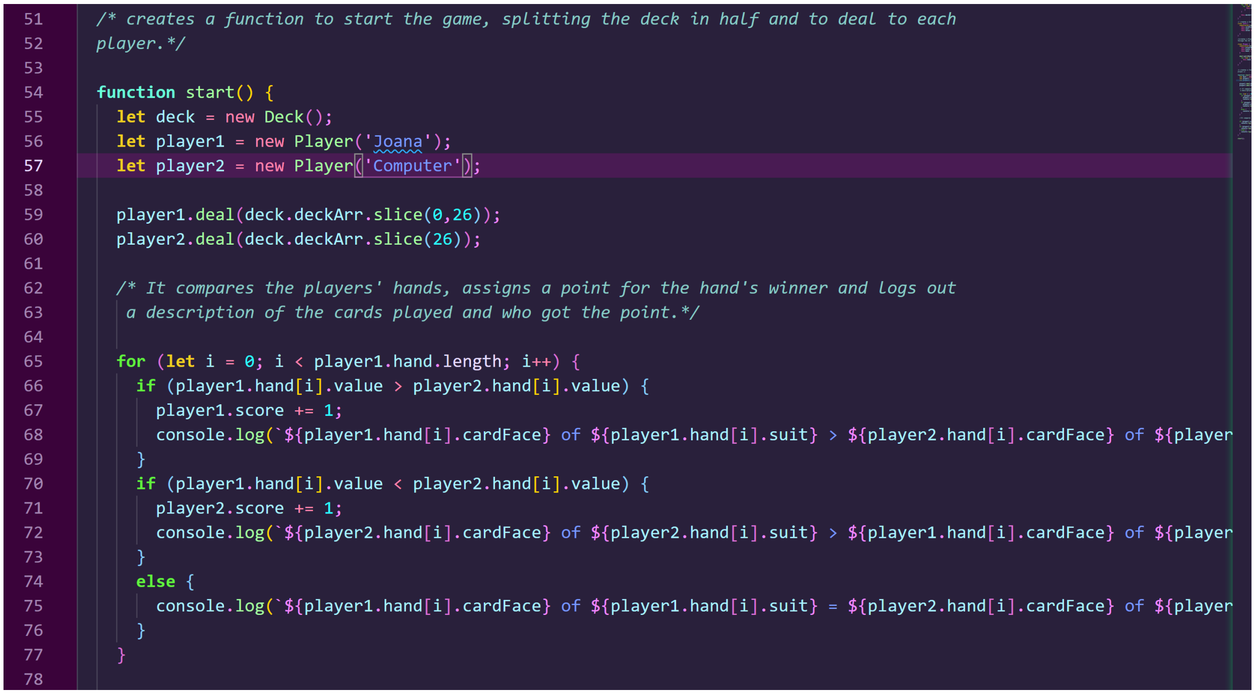
Screenshots of Code:



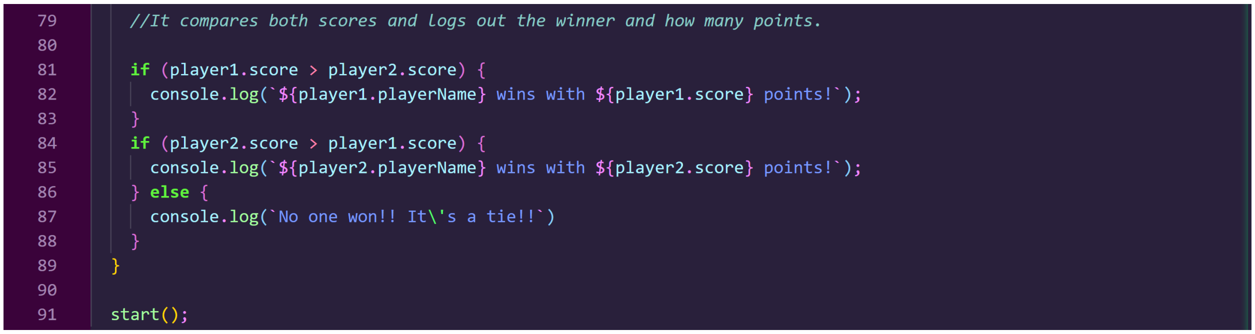
```
/*creates a Deck class, that contains a deck array with all elements that a deck has.
 It also goes through the cardFaces, Values and Ranks of those cards to be returned
into that array.
It sorts them as well in a random way so they are shuffled.*/
  class Deck {
    constructor() {
      this.deckArr = [];

      const cardFaces = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K'];
      const suits = ['Clubs', 'Hearts', 'Spades', 'Diamonds' ];
      const values = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];

      for (let suit in suits) {
        for (let cardFace in cardFaces) {
          this.deckArr.push(new Card(cardFaces[cardFace], suits[suit], values[cardFace]));
        }
      }

      this.deckArr.sort(() => Math.random() - 0.5);
    }
  }

```



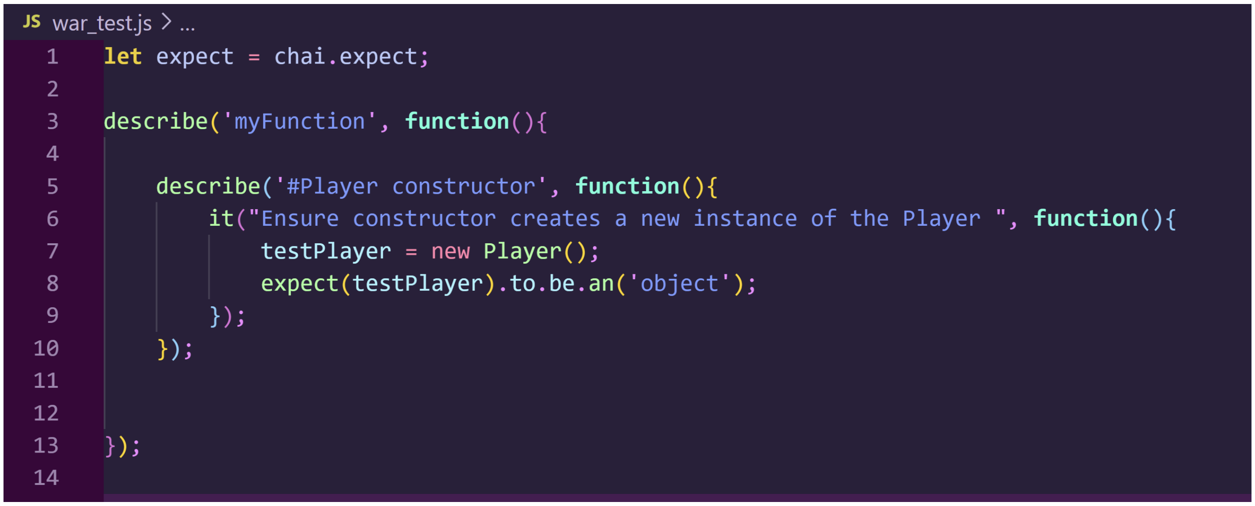
```
// creates a Card class with the necessary properties related to each card.
class Card {
  constructor(cardFace, suit, value, ) {
    this.cardFace = cardFace;
    this.suit = suit;
    this.value = value;


  }
}

/*creates a Player class, with the necessary properties and creates a method to loop
through the 26 cards for each player.*/

class Player {
  constructor(playerName) {
    this.playerName = playerName;
    this.score = 0;
    this.hand = [];
  }

  deal(splitDeck){
    for (let i = 0; i < 26; i++) {
      this.hand.push(splitDeck[i]);
    }
  }
}

```



```
/* creates a function to start the game, splitting the deck in half and to deal to each
player.*/

function start() {
  let deck = new Deck();
  let player1 = new Player('Joana');
  let player2 = new Player('Computer');

  player1.deal(deck.deckArr.slice(0,26));
  player2.deal(deck.deckArr.slice(26));

  /* It compares the players' hands, assigns a point for the hand's winner and logs out
   a description of the cards played and who got the point.*/

  for (let i = 0; i < player1.hand.length; i++) {
    if (player1.hand[i].value > player2.hand[i].value) {
      player1.score += 1;
      console.log(`${player1.hand[i].cardFace} of ${player1.hand[i].suit} > ${player2.hand[i].cardFace} of ${player
    }
    if (player1.hand[i].value < player2.hand[i].value) {
      player2.score += 1;
      console.log(`${player2.hand[i].cardFace} of ${player2.hand[i].suit} > ${player1.hand[i].cardFace} of ${player
    }
    else {
      console.log(`${player1.hand[i].cardFace} of ${player1.hand[i].suit} = ${player2.hand[i].cardFace} of ${player
    }
  }

```



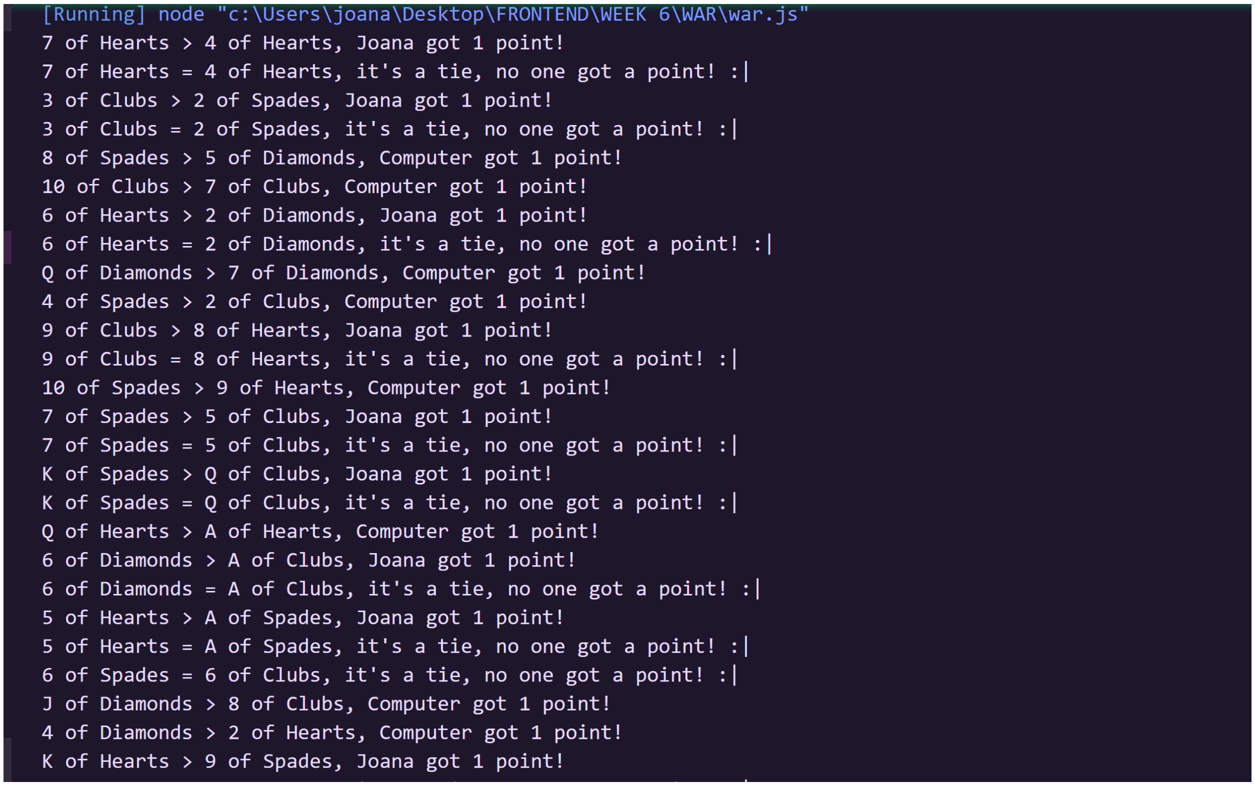
```
  //It compares both scores and logs out the winner and how many points.

  if (player1.score > player2.score) {
    console.log(`${player1.playerName} wins with ${player1.score} points!`);
  }
  if (player2.score > player1.score) {
    console.log(`${player2.playerName} wins with ${player2.score} points!`);
  } else {
    console.log(`No one won!! It\'s a tie!!`)
  }
}

start();
```



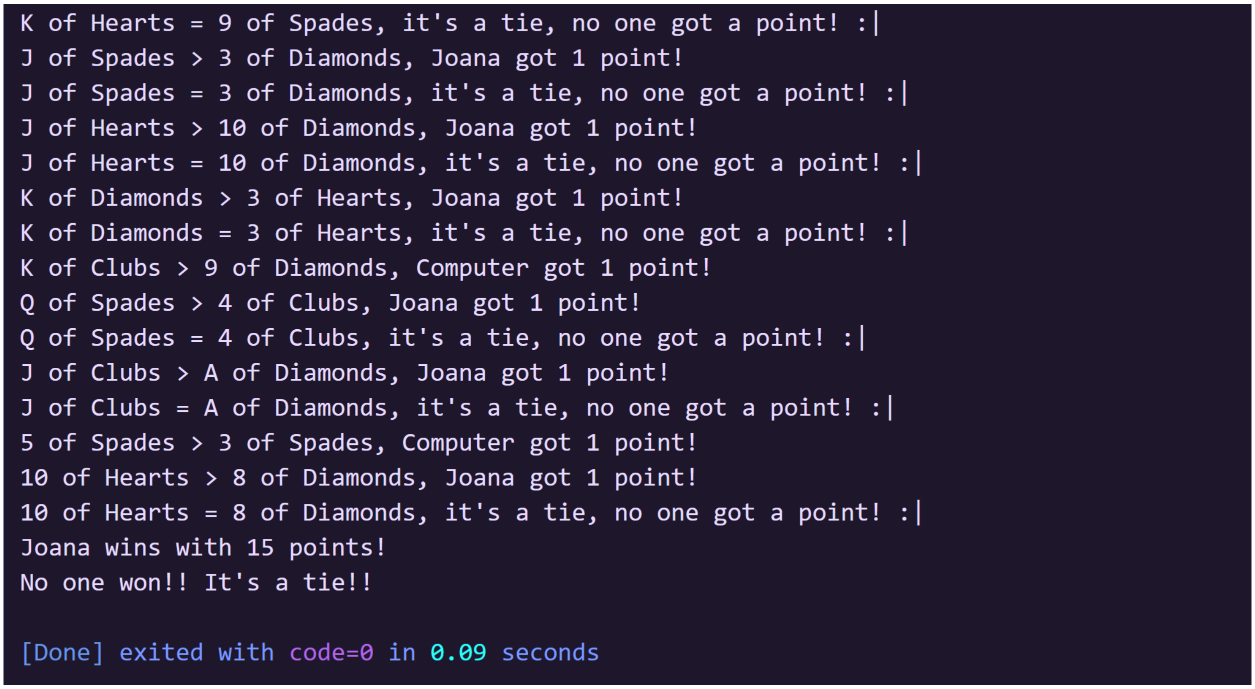
JS war_test.js > ...

```
let expect = chai.expect;

describe('myFunction', function(){

    describe('#Player constructor', function(){
        it("Ensure constructor creates a new instance of the Player ", function(){
            testPlayer = new Player();
            expect(testPlayer).to.be.an('object');
        });
    });


});

```

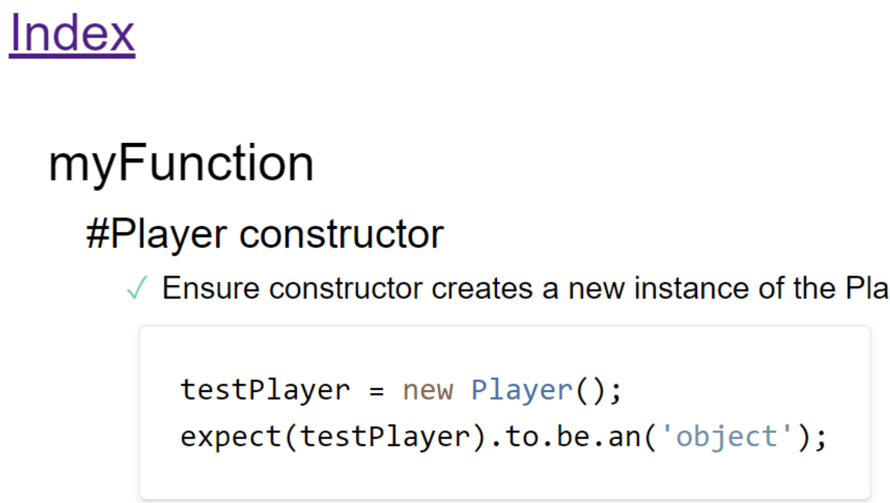
Screenshots of Running Application:



```
[Running] node "c:\Users\joana\Desktop\FRONTEND\WEEK 6\WAR\war.js"
7 of Hearts > 4 of Hearts, Joana got 1 point!
7 of Hearts = 4 of Hearts, it's a tie, no one got a point! :|
3 of Clubs > 2 of Spades, Joana got 1 point!
3 of Clubs = 2 of Spades, it's a tie, no one got a point! :|
8 of Spades > 5 of Diamonds, Computer got 1 point!
10 of Clubs > 7 of Clubs, Computer got 1 point!
6 of Hearts > 2 of Diamonds, Joana got 1 point!
6 of Hearts = 2 of Diamonds, it's a tie, no one got a point! :|
Q of Diamonds > 7 of Diamonds, Computer got 1 point!
4 of Spades > 2 of Clubs, Computer got 1 point!
9 of Clubs > 8 of Hearts, Joana got 1 point!
9 of Clubs = 8 of Hearts, it's a tie, no one got a point! :|
10 of Spades > 9 of Hearts, Computer got 1 point!
7 of Spades > 5 of Clubs, Joana got 1 point!
7 of Spades = 5 of Clubs, it's a tie, no one got a point! :|
K of Spades > Q of Clubs, Joana got 1 point!
K of Spades = Q of Clubs, it's a tie, no one got a point! :|
Q of Hearts > A of Hearts, Computer got 1 point!
6 of Diamonds > A of Clubs, Joana got 1 point!
6 of Diamonds = A of Clubs, it's a tie, no one got a point! :|
5 of Hearts > A of Spades, Joana got 1 point!
5 of Hearts = A of Spades, it's a tie, no one got a point! :|
6 of Spades = 6 of Clubs, it's a tie, no one got a point! :|
J of Diamonds > 8 of Clubs, Computer got 1 point!
4 of Diamonds > 2 of Hearts, Computer got 1 point!
K of Hearts > 9 of Spades, Joana got 1 point!
```



```
K of Hearts = 9 of Spades, it's a tie, no one got a point! :|
J of Spades > 3 of Diamonds, Joana got 1 point!
J of Spades = 3 of Diamonds, it's a tie, no one got a point! :|
J of Hearts > 10 of Diamonds, Joana got 1 point!
J of Hearts = 10 of Diamonds, it's a tie, no one got a point! :|
K of Diamonds > 3 of Hearts, Joana got 1 point!
K of Diamonds = 3 of Hearts, it's a tie, no one got a point! :|
K of Clubs > 9 of Diamonds, Computer got 1 point!
Q of Spades > 4 of Clubs, Joana got 1 point!
Q of Spades = 4 of Clubs, it's a tie, no one got a point! :|
J of Clubs > A of Diamonds, Joana got 1 point!
J of Clubs = A of Diamonds, it's a tie, no one got a point! :|
5 of Spades > 3 of Spades, Computer got 1 point!
10 of Hearts > 8 of Diamonds, Joana got 1 point!
10 of Hearts = 8 of Diamonds, it's a tie, no one got a point! :|
Joana wins with 15 points!
No one won!! It's a tie!!

[Done] exited with code=0 in 0.09 seconds
```



Index

myFunction

#Player constructor

✓ Ensure constructor creates a new instance of the Player

```
testPlayer = new Player();
expect(testPlayer).to.be.an('object');
```

URL to GitHub Repository:

https://github.com/JoanaBarao7/Week6-WarCarGame