# 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.* 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
  + 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:**

const SUITS =["♥","♦","♠","♣"]

const VALUES=["A","2","3","4","5","6","7","8","9","10","J","Q","K"]

const REALValues = {

    "2":2,

    "3":3,

    "4":4,

    "5":5,

    "6":6,

    "7":7,

    "8":8,

    "9":9,

    "10":10,

    "J":11,

    "Q":12,

    "K":13,

    "A":14,

}

class Deck{

    constructor(cards)

    {

        this.cards=cards;

    }

}

class Card{

    constructor(suit,value){

        this.suit=suit;

        this.value=value;

    }

}

const Player1 = [];

    const Player2=[];

function FullDeck()

{

    let Deck=[];

    VALUES.forEach( value =>{

        SUITS.map(suit => {

            Deck.push({suit,value})

        })

    })

    while(Deck.length > 0){

        const rand1 =Math.floor(Math.random() \* Deck.length);

        const getItem1= Deck.splice(rand1,1);

        const rand2= Math.floor(Math.random() \* Deck.length);

        const getItem2= Deck.splice(rand2,1);

        Player1.push(getItem1);

        Player2.push(getItem2);

    }

    return [Player1.flat() ,Player2.flat()]

}

//let array;

//function Shuffle(array){

function RoundWinner()

{

FullDeck()

let CardOne =[];

let CardTwo = [];

CardOne=Player1.flat()

CardTwo=Player2.flat()

let CardOneCounter = 0;

let CardTwoCounter = 0;

for(let i=0; i < CardOne.length && i < CardTwo.length ; i++)

{

if ([CardOne[i].value] > [CardTwo[i].value])

{

    CardOneCounter = CardOneCounter+1;

    console.log("Player One Card is " + (CardOne[i].value) + " Player Two Card is " + (CardTwo[i].value))

    console.log("PlayerOne Won Round "+ i +  " Card Value equal  " + REALValues [CardOne[i].value])

}else if([CardOne[i].value] < [CardTwo[i].value])

{

    console.log("Player One Card is " + (CardOne[i].value) + " Player Two Card is " + (CardTwo[i].value))

    console.log("PlayerTwo Won Round " +i + " Card Value equal  "+ REALValues [CardTwo[i].value])

    CardTwoCounter = CardTwoCounter +1;

}

}

if (CardOneCounter > CardTwoCounter){

    console.log("Player One  Won with "+ CardOneCounter + " Rounds Vs " + CardTwoCounter + " For Player Two" )

}else {

    console.log("Player Two  Won with "+ CardTwoCounter + " Rounds Vs" + CardOneCounter + " For Player One" )

}

 console.log(CardOneCounter)

 console.log(CardTwoCounter)

}

console.log(RoundWinner());

nit Testing

  describe ("myFunction" ,() => {

        describe("#FullDeck", () => {

            it('Should shuffle the whole Cards into 26 cards to each player ', () => {

                const game = FullDeck()

                expect (Player1 != Player2)

        });

    });

})

HTML file

<!DOCTYPE html>

<html>

<head>

 <link rel="stylesheet" href="<https://cdn.rawgit.com/mochajs/mocha/2.2.5/mocha.css>">

</head>

<body>

<div id="mocha"><p><a href=".">Index</a></p></div>

<div id="messages"></div>

<dev id="fixtrues"></div>

<script src=<https://cdn.rawgit.com/mochajs/mocha/2.2.5/mocha.js>></script>

<script src=<https://cdn.rawgit.com/chaijs/chai/3.5.0/chai.js>></script>

<script src="WarGame.js"></script>

<script> src=mocha.setup('bdd');</script>

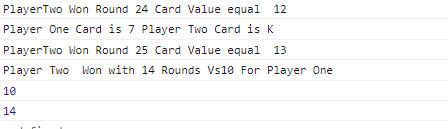
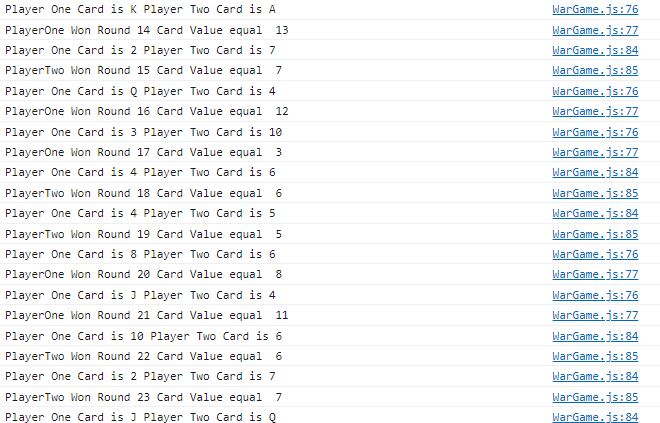
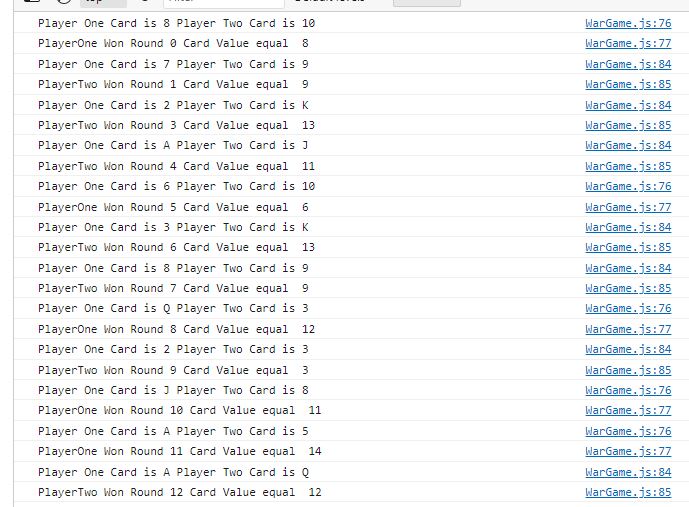
<script src="TestCase.js"></script>

<script> mocha.run(); </script>

</body>

</html>

**Screenshots of Running Application:**

****

**HTML-Unit Testing**

****

**URL to GitHub Repository**