

# COMP2132 :: ASSIGNMENT 04

**OBJECTIVE** Create and use custom Javascript Objects

**REQUIREMENTS** Use the file named **assignment04\_start.js** (provided with this .pdf) to get started with this assignment.

This assignment can be completed in three stages. It is recommended students complete stage 1 before proceeding to the 2nd stage, and completing that before stage 3.

## Stage 1

Define a **constructor function** or **class** for representing a playing **Card**. The constructor must require the following three parameters:

- **face** eg: "9", "10", "Jack"
- **value** eg: 9, 10, 10 (an ace would get a value of 1)
- **suit** eg: "Diamond", "Spade"

Include with the Card Object a function called **describeSelf()** that will return a string description following this format: **"FACE of SUITS. Value: VALUE"** eg: **"King of Hearts. Value: 10"**

Instantiate a single Card object for a King of Hearts, invoke the **describeSelf()** function on the Card and display the results to the browser.

## Stage 2

Update the constructor that has already been defined for the Deck class so that it will instantiate the 52 Cards Objects required to populate a Deck. Due to the limitations of function scope, you may need to use nested **for** loops (instead of nested **forEach** loops) with the arrays already defined for you:

- `this.faces = ["Ace", 2, 3, 4, 5, 6, 7, 8, 9, 10, "Jack", "Queen", "King"];`
- `this.values = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10];`
- `this.suits = ["Spade", "Club", "Heart", "Diamond"];`

Each new Card Object instantiated should be added to the Deck variable array named **cards** using the `Array.push()` function.

Instantiate a new Deck, **shuffle()** the cards and run several rounds of **dealCard()** and **describeSelf()** functions, displaying the results to the web browser.

**NOTE:** For stage 2, other than updating the code inside the Deck constructor function, make no changes to the Deck code (the prototype.functions do not require any changes)

### Stage 3

Define a **constructor function** or **class** for representing a **Player** object. The constructor must require the following parameter:

- **name** eg: "Shania", "Joe"

The constructor for the Player object should initialize an empty array for storing one or more Cards. This array will represent the Player's hand in a card game.

Include with the Player object functions for the following:

- **addCardToHand( Card )** this function requires a Card Object as parameter and will use `Array.push()` to add the Card Object to the Player's hand
- **describeSelf( )** this function should return a string that includes the Player's name and an HTML list of all the Cards in the Player's hand

Instantiate two new Player Objects, giving each a different name. Instantiate a new **Deck** object and **shuffle( )** it. Use **dealCard( )** as the parameter value for **addCardToHand( )** several times to deal five cards to each player. Invoke the **describeSelf( )** functions of both players and display the results to the browser.

---

**BONUS** Additional challenge, **NOT REQUIRED**: Change the **describeSelf()** functions of Card Objects to display an appropriate card image instead of plain text. For example, instead of displaying the text description "Eight of Hearts, Value: 8" display an image of the a playing card and instead insert the text description as the value of the `img` tag's `alt` attribute. See **card\_images/** provided with this assignment for a collection of suitable images to use.

Note: No additional marks for including the bonus challenge. Do NOT attempt unless essential requirements have already been met.

---

**SUBMISSION** **Assigned:** Session 5

**Due:** 11:59pm the night before Session 6

Compress all files (HTML, CSS, JS, etc) and folders into a single **.ZIP** file and upload it to the Drop Box: <https://learn.bcit.ca> > COMP2132 > Session05 > Assignment04 Drop Box  
Assignments must be uploaded to the Drop Box before the due date.

---