Javascript Assignment 10

1. Are Higher Order functions and Call back functions the same ? If not, briefly explain about both functions.
   Ans : "**Higher Order functions**" are the functions that accepts functions as parameter and/or return a function.
   Whereas **Callback function** is the function that is passed as a parameter to a function.

   function doSomething(var1, var2, myFunc){
           //do some operations using var1 and var2.
           let output = var1 + var2;

           myFunc(output);
   }

   In above code **doSomething() is higher order function** as its taking myFunc as parameter.

   myFunc() is callback function as its passed as a parameter to doSomething().

2. Is filter a Higher Order function in Javascript ? If yes, why ?
   Ans: YES. filter() is an higher order function as it takes a predicate function as argument.

   Example :

```javascript
let input = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18];

let even = input.filter((num)=>{
    if(num%2==0){
        return true;
    }else{
        return false;
    }
});

console.log(even);
```

3. Give an example of a Higher Order function and a call back function
   used in the same program.
   <mark>Ans:</mark>

```javascript
function display(data){
    console.log(data);
}

function addAndPrint(num1, num2, printOutput){
    let sum = num1 + num2;
    printOutput(sum);
}

let num1 = 10;
let num2 = 20;

addAndPrint(num1, num2, display);
```

<mark>In above example. **addAndPrint**() is **higher order** function and **printOutput() is**</mark>
<mark>**callback** function.</mark>

4. Carefully check the example below:
   a) What will be the output of this program ?
   <mark>Ans: Output will be : -</mark>

   <mark>Hello John</mark>
   <mark>Hello Tina</mark>
   <mark>Hello Kale</mark>
   <mark>Hello Max</mark>

   b) Which function is a Higher Order function here ?
   <mark>Ans: **useFunction**() is higher order function</mark>

   ```javascript
   const names= ['John', 'Tina','Kale','Max']
   function useFunction(arr,fn){
   for(let i=0; i<arr.length; i++){
   fn(arr[l]);
   }
   ```

```
}
function argFn (name){
console.log("Hello " + name );
}
useFunction(names,argFn);
```