**Assignment 10**

**1.**Are Higher Order functions and Call back functions the same ? If not, briefly explain about both functions?

Higher order functions and Call back functions are different functions. Higher Order Functions is defined as the functions that use other functions as arguments or return function from it. In other words, the outer function that takes in a function is called as higher order functions.

Callback function is a function passed into another function as an argument, which is then invoked to complete some sort of action.

function higherorderfunction(callback)

{

//do something

callback(); //invoke the callback

}

callback(){ //steps to do }

Example program:

//handle specifies what to do with the result after adding a and b

function add(a,b,handle)

{

    let result=a+b;

    handle(result);

}

function print(result)

{

    console.log("printing from result:",result)

}

function sendReq(result)

{

    console.log("sending data to server:",result)

}

function nothingdoingwithresult(result)

{

    console.log(result);

}

   add(4,5,print)   //output: Printing from result:9

add(4,5,sendReq)    //output: sending data to server: 9

  add(4,5,nothingdoingwithresult)

//in this func nothingdoingwithresult..it will simply print the result

Output:

printing from result: 9

sending data to server: 9

9

In the above example, handle is a call back function and ‘add’ in add(a,b,handle) is a higher order function. Higher order function takes in call back function and call back function is called when required.

2. Is filter a Higher Order function in Javascript ? If yes, why ?

Filter is a built in Higher order function in javascript. The filter function takes in a call back that returns either true or false value.

let numbers=[1,2,3,4]

let evennum=numbers.filter(x=> x%2==0)

console.log(evennum)

3. Give an example of a Higher Order function and a call back function used in the same program.

myarr=[1,2,4]

function calculate(operation, initialValue, numbers) {

  let total = initialValue;

  for (const number of numbers) {

    total = operation(total, number);

  }

  return total;

}

function sum(n1, n2) {

  return n1 + n2;

}

function multiply(n1, n2) {

  return n1 \* n2;

}

// console.log(calculate(sum, 0, [1, 2, 4]));      // => 7

// calculate(multiply, 1, [1, 2, 4]); // => 8

console.log(calculate(multiply,1,myarr));

console.log(calculate(sum,0,myarr));

Higher Order function and Callback functions implemented in the same program.

calculate is a higher order function. Sum()and multiply() is a callback function.

4. Carefully check the example below:

a) What will be the output of this program ?

const names=['John','Tina','kale','Max']

function useFunction(arr,fn)

{

    for(let i=0;i<arr.length;i++)

    {

       fn(arr[i])

    }

}

function argFn(name)

{

    console.log("Hello "+ name)

}

useFunction(names,argFn)

4a) output:

Hello John

Hello Tina

Hello kale

Hello Max

The array “names” is passed as arr in useFunction. For loop is used to get array of names and passed within the function argFn(name)…Hello and each name is printed..

b) Which function is a Higher Order function here ?

**useFunction(names,argFn)** is a Higher Order Function..this functions that use other functions as arguments.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*