**19th January**

**Higher order function** is a function that accepts functions as parameters and/or returns a function. They can be passed into other functions as parameters or returned from them as well. Higher-order functions are only possible because of the First-class function.

Functions such as[filter()](https://www.geeksforgeeks.org/javascript-array-filter-method/), [map()](https://www.geeksforgeeks.org/javascript-array-map-method/), [reduce(),](https://www.geeksforgeeks.org/javascript-array-reduce-method/)[some(),](https://www.geeksforgeeks.org/javascript-array-some-method/) etc, all are examples of Higher-Order Functions.

**FILTER() :-**

Filter, is a method that helps to filter the array. The filter function takes in a call back that returns either true or false value. Those items in the array that return the true value on the test condition will be the element of the new filtered array.

Example: -

//filter positive numbers

const numArray = [-2, 1, 50, 20, -47, -20]

const positiveArray = numArray.filter((num) => {

return num > 0

})

console.log(positiveArray)

//output- [1, 50, 45, 20]

**MAP() :-**

The map method called on an array will return the array of the **same size**. The returned array can be the **result of the operation you apply on** the array call the method on.

Example: -

//Extract the marks of students

const studentsArray = [

{

name: 'aryan',

marks: 90

},

{

name: 'kamal',

marks: 95

},

{

name: 'malik',

marks: 6

}

]

const scoreArray = studentsArray.map((student) => {

return students.marks

})

console.log(scoreArray)

//output- [90, 95, 6]

**REDUCE() :-**

The reduce() basically does is start with a collection and a variable with an initial value. It then iterate over the collection and append (or add) the values to the variable.

Example:-

//a basic example of reduce()

const numArray = [1, 2, 3, 4, 5]

const sum = numArray.reduce((total, num) => {

return total + num

})

console.log(sum)

//output- 15