ES6 Assignments

1. **Symbols:**Write a class that defines next() method to return next number from Fibonacci series. The class will have a private attributes ‘previousNo’ & ‘currentNo’.

class MyClass {

  private previousNo: number;

  private currentNo: number;

  constructor(curr: number) {

    this.previousNo = fib(curr - 1);

    this.currentNo = curr;

  }

  next() {

    return fib(this.currentNo + 1);

  }

}

var fib = function (n: number) {

  var a = 1;

    for(var i=2;i<=n;i++){

     a=a\*i;

    }

  return a;

};

let myClass = new MyClass(5);

console.log(myClass.next());

1. **Iterators:**Write a program that returns next Armstrong number after calling getNextArmstrong() method.

let n = 10;

function armstong() {

  for(let i=n;i>0;i++) {

    let digits = i.toString().length;

    let sum = 0;

    let s = i.toString();

    for (let c of s) {

      sum += Math.pow(parseInt(c), digits);}

    if (sum == i) {

      console.log(i);

      n = i;

      break;}}}

function getNextArmstrong() {

  n++;

  armstong();}

getNextArmstrong();

getNextArmstrong();

getNextArmstrong();

1. **Generators:**Write a program that returns next Armstrong number after calling getNextArmstrong() method. Add functionality to reset generating Armstrong number from zero. In case, Armstrong number goes above one thousand then throw an error.

let n = 10;

function armstrong() {

  if(n<1000){

  for(let i=n;i>0;i++) {

    let digits = i.toString().length;

    let sum = 0;

    let s = i.toString();

    for (let c of s) {

      sum += Math.pow(parseInt(c), digits);

    }

    if (sum == i) {

      console.log(i);

      n = i;

      break;

    }

  }}else{

    reset();

  }

}

function getNextArmstrong() {

  n++;

  armstrong();

}

function reset() {

  n = 10;

}

getNextArmstrong();

getNextArmstrong();

getNextArmstrong();

getNextArmstrong();

getNextArmstrong();

getNextArmstrong();

getNextArmstrong();

1. **Collections:** Using Set & Map, create a static data for chatting application. Here we have 2 chatrooms, every chatroom is having 3 users & every user has posted different messages in a chat room. Note that one user can belong to a single chat room only. Now you need to find out how you will hold this data using Set & Map data structures. Also add functionality to get list of all users from a specific chatroom & listing down all message from a chatroom.