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.
2. class User {
3. name: string;
4. message: Map<string, string>;
5. constructor(name: string, message: Map<string, string>) {
6. this.name = name;
7. this.message = message;
8. }
9. }
10. let msgUser1 = new Map();
11. msgUser1.set("message1", "This is the message 1");
12. msgUser1.set("message2", "This is the message 2");
13. let user1 = new User("PK", msgUser1);
14. let msgUser2 = new Map();
15. msgUser2.set("message1", "This is the message 1");
16. msgUser2.set("message2", "This is the message 2");
17. let user2 = new User("SK", msgUser2);
18. let msgUser3 = new Map();
19. msgUser3.set("message1", "This is the message 1");
20. msgUser3.set("message2", "This is the message 2");
21. let user3 = new User("RK", msgUser3);
22. let msgUser4 = new Map();
23. msgUser4.set("message1", "This is the message 1");
24. msgUser4.set("message2", "This is the message 2");
25. let user4 = new User("KJ", msgUser4);
26. let msgUser5 = new Map();
27. msgUser5.set("message1", "This is the message 1");
28. msgUser5.set("message2", "This is the message 2");
29. let user5 = new User("AY", msgUser5);
30. let msgUser6 = new Map();
31. msgUser6.set("message1", "This is the message 1");
32. msgUser6.set("message2", "This is the message 2");
33. let user6 = new User("RP", msgUser6);
34. let usersSet = new Set<User>();
35. let usersSet2 = new Set<User>();
36. usersSet.add(user1);
37. usersSet.add(user2);
38. usersSet.add(user3);
39. function addUserRoom2(user: User) {
40. if (!usersSet.has(user) && usersSet2.size < 3) {
41. usersSet2.add(user);
42. }
43. }
44. addUserRoom2(user4);
45. addUserRoom2(user5);
46. addUserRoom2(user1);
47. addUserRoom2(user6);
48. let chatroom = new Map<String, Set<User>>();
49. chatroom.set("chatroom1", usersSet);
50. chatroom.set("chatroom2", usersSet2);
51. for (let value of chatroom.values()) {
52. for (let key of value.keys()) {
53. console.log(key);
54. }
55. }