ES6 Assignments – A1

1. **Constants:** Declare a constant & confirm its value cannot be changed.

Ans:-

```
"use strict";
var colors = ['red'];
colors.push('green');
console.log(colors); // ["red", "green"]
colors.pop();
colors.pop();
console.log(colors); // []
colors = []; // TypeError
```

```
> node "c:\Users\ASUS\Desktop\ES2015\Scripts\Constants.js"
:\Users\ASUS\Desktop\ES2015\Scripts\Constants.js"
[ 'red', 'green' ]
[]
```

2. **Scoping:** Declare a variable inside if condition & make sure that it is not accessible outside if condition

Ans:-

```
"use strict";
function forScope() {
    for (var i = 0; i < 5; i++) {
        console.log(i); //prints the values 0 through 4;
    }
    console.log(i); //prints 5;
}
forScope();
```

```
> node "c:\Users\ASUS\Desktop\ES2015\Scripts\Scoping.js"
0
1
2
3
4
5
```

3. **Enhanced object properties:** Create an 'Order' object having data members 'id', 'title', 'price'. Add the methods printOrder() & getPrice(). Now, copy the order object using Object.assign().

Ans:-

```
function Order  (id, title, price){
    this.id = id;
    this.title = title;
    this.price = price;
 }

 order1 = new Order( 1 , "Bhsuahn Bire"      , 100 ) ;
 order2 = new Order() ;
 var order2    = Object .assign(order1) ;

 getPrice  = function(){
    console. log ( order1  .price) ;
 }();

 printOrder  = function  (){
    console. log ( order1 .id  +"  "+ order1 .title + " " + order1.price ) ;
    console. log ( order2 .id  +"  "+ order2 .title + "  " + order2.price ) ;
}();
```

```
> node "c:\Users\ASUS\Desktop\ES2015\Scripts\Enhnaced Obj.js"
100
1  Bhsuahn Bire 100
1  Bhsuahn Bire  100
```

4. **Arrow functions:** Take an array of strings & convert it into another array of object which has two properties (string, string_length). For example.

Ans:-

```
let names = [" Tom", " Ivan", " Jerry"]
let arrObject = ( names) => {
let arr = [ ];
for ( let n of names) {
arr. push( {name: n, length: n.length} );
}
console.log( arr );
}
arrObject(names);
```

```
> node
"c:\Users\ASUS\Desktop\ES2015\Scripts\Arrow Func.js"
[
  { name: ' Tom', length: 4 },
  { name: ' Ivan', length: 5 },
  { name: ' Jerry', length: 6 }
]
```

5. Extended parameter handling:

a. Write a add() with default values.

Ans:-

```
function add( a = 10, b = 5 ) {
    return a + b ;
    }
console.log(add());
```

```
> node
"c:\Users\ASUS\Desktop\ES2015\Scripts\Extended Parameter Handling (a).js"
15
```

b. Write a function user Friends() that takes 2 arguments username & array of user friends. The function should print username & his list of friends. (Use rest parameters).

Ans:-

```
function userFriends( username, ...args) {
    return username+ " ,"+ args;
}
console.log(userFriends( " abc", " xyz", " pqr") );
```

```
> node
"c:\Users\ASUS\Desktop\ES2015\Scripts\Extended Parameter Handling (b).js"
 abc , xyz, pqr
```

c. Write a function printCapitalNames() that takes five names as argument & prints them in capital letters. Use spread operator in order to call printCapitalNames() function.

Ans:-

```
const names = [' Ali', ' Atta', ' Alex', ' John', ' Amy'] ;
const printCapitalNames = names. map( name =>
name.toUpperCase());
console.log( printCapitalNames)
```

```
> node "c:\Users\ASUS\Desktop\ES2015\Scripts\Extended Parameter Handling (c).js"
[ ' ALI', ' ATTA', ' ALEX', ' JOHN', ' AMY' ]
```

6. Template literals:- Draft a ticket to Sysnet that describes problem with your laptop. Use 'template literals' to add value of laptop model, your desk no, your name etc.

Ans:-

```
let laptop_model = "Asus VivoBook";
let desk_no = "515";
let name = "Bhushan Bire";
```

```
console.log( `I am ${name} have some issue with
${laptop_model} and my desknumber is ${desk_no}.`);
```

> I am Bhushan Bire have some issue with Asus VivoBook and my desk number is 515.

7. De-structuring assignment:

a. Suppose there is a javascript array with 4 elements. Print the value of 3rd element using array matching.

Ans:-

```
Array.prototype.diff = function(arr) {
    var ret = [ ];
    for(var i in this) {
        if( arr.indexOf( this[i] ) > - 1 ) {
            ret.push( this[i] );
        }
    }
    return ret;
};

var array1 = [ "Hey", "I", "am", "Bhushan"] ;
var array2 = [ "Do" , "you", "know", "Javascript"] ;
console.log(array1.diff(array2));
```

```
> node "c:\Users\ASUS\Desktop\ES2015\Scripts\De-
structuring(a).js":\Users\ASUS\Desktop\ES2015\Scripts\De-structuring(a).js"
:\Users\ASUS\Desktop\ES2015\Scripts\De-structuring(a).js"
[am]
```

b. Create an organization object having attributes name, address. Write a program to retrieve pin code of an address using object deep matching.

Ans:-

```
let organisation1 = {
        name: " John",
        address: {
        street: " Rammurthy n agar",
        city: " Bangalore",
        pincode: 560016
    }
}
    let organisation2 = {
        name: " Robert",
        address: organisation1. address
    }
```

```
console. log( (organisation1. address. pincode === organisation2. address. pincode) );
console. log( organisation1. name === organisation2. name) ;
```

```
> node "c:\Users\ASUS\Desktop\ES2015\Scripts\De-structuring(b).js"
true
false
```

8. **Classes & Modules:** Write a class Account with attributes id, name, balance. Add two subclasses SavingAccount & CurrentAccount having specific attribute interest & cash credit respectively. Create multiple saving & current account objects. Write functionality to find out the total balance in the bank.

Ans:-

```
class Account {
    constructor( id, name, balance) {
    this. id = id;
    this. name = name;
    this. balance = balance;
    }
    }
    class SavingsAccount extends Account {
    constructor( id, name, balance, interest) {
    super( id, name, balance) ;
    this. interest = interest;
    }
    totalBalance( ) {
    let newBalance = this. balance * this. interest;
    this.balance = this.balance + newBalance;
    return this.balance
    }
    }
    class CurrentAccount extends Account {
    constructor( id, name, balance, cash_credit) {
    super( id, name, balance) ;
    this.cash_credit = cash_credit;
    }
    totalBalance( ) {
    let newBalance = this.balance * this.cash_credit;
    this.balance = this.balance + newBalance;
    return this. balance;
    }
    }
    var saving = new SavingsAccount( " 987612345128", "Bhushan", 200000, 1.5) ;
    var current = new CurrentAccount( " 987612345129", "Nikhil", 500000, 0.5) ;
    console.log( saving.totalBalance( ));
    console.log( current.totalBalance( ));
```

```
500000

750000
```