Assignment 1

-Gaurav Kumar Singh

1)

const a = 7;

console.log(a);

a = 5; //SyntaxError: Babel: "a" is read-only

2)

let a = 7;

if(a == 7){

  let b = 4;

}

else

  {

    b = 5;

  }

console.log(b);//b is not defined

3)

const order = { id: '1',

title: 'pizza',

price: 700,

printOrder: function(){return this.title;},

getPrice: function(){return this.price;}

};

const newOrder = Object.assign({}, order);

console.log(newOrder);

4)

let names = ["Tom", "Ivan", "Jerry"]

const newArray = names.map((str) => ({name: str, length:str.length}))

console.log(newArray);

5)

a)

function add(a = 7, b= 7){

  return a+b;

}

const sum = add();

console.log(sum);

b)

let friends = ["Xander", "Blake", "Elijah"]

function userFriends(name1, ...friend)

{

  console.log(name1);

  for(let i in friend)

  {

    console.log(friend[i]);

  }

}

let name1 = "Noah";

userFriends(name1, friends);

c)

let names = ["xander", "blake", "noah", "ashton", "philip"]

function printCapitalNames(...namess)

{

  for(let i in namess)

  {

    console.log(namess[i].toUpperCase());

  }

}

printCapitalNames(...names);

6)

function description(lModel,desk\_no,name) {

let text = {

"model":lModel,

"no":desk\_no,

"name":name

}

console.log(`This is the Model : ${text.model}\n

This is the NO : ${text.no}\n

This is the Name : ${text.name}`);

}

description("EA12",12,"DELL");

7)

let array = [1,2,3,4,];

let [ , , third , ] = array;

console.log(third);

const surr = {

"name":"roy",

"address":"here",

"pincode":1234

};

let { , , pincode} = surr;

console.log(pincode);

8)

class Accounts{

private total\_Balance;

constructor(balance){

this.total\_Balance = balance;

}

get Balance(){

return this.total\_Balance;

}

deposit(amount){

this.total\_Balance += amount;

console.log("Deposited : $"+ amount);

console.log("Total Balance : $"+ this.Balance);

}

withdraw(amount){

if (amount < this.Balance) {

this.total\_Balance -= amount;

console.log("Withdraw Amount : $"+ amount);

console.log("Remaining Balance : $"+ this.Balance);

} else {

return "Less Cash Present";

}

}

}