# ES6 Exercises

## 1. Scope & Context in JS

(i) What will be the output

```
function Bar(){
  (function self(){
  console.log(this);
  })();
}
function Drinks(){
(()=> console.log(this))();
}
const barObj = new Bar();
const drinksObj = new Drinks();
```

Window Object

Function Object

(ii) What will be the output

```
let foo = 0;
function bar() {
        if (!foo) {
                let foo = 10;
                console.log(foo);
        }
        console.log(foo);
}
bar();
```

Outout: 10, 0

(iii) What will be the output

```
let  guessMe1 = 1;
let guessMe2 = 2;
```

```
{
   try {
      console.log( guessMe1, guessMe2 ); // (A)
   }
   catch (err){
         console.log("Oops", err)
   }
   let guessMe2 = 3;
   console.log( guessMe1, guessMe2 ); // (B)
}
console.log( guessMe1, guessMe2 ); // (C)
```

guessMe2 not defined

1, 3

1, 2

## 2. Arrow Function

(i) What will be the output

```
const  myfunc = (list) => arguments[0].sort();
const myList= myfunc([10,20,25]);
console.log(myList);
```

Error: Arguments not defined in arrow function

(ii) What will be the output

```
function foo(n) {
const f = () => arguments[0] + n;
return f();
}
const myfoo = foo(1);
console.log( myfoo );
```

Ans: 2, Coz arg taking parents object prms

(iii) What will be the output

```
var obj = {
i: 10,
b: () => console.log(this.i, this),
```

```
c: function() {
console.log(this.i, this);
}
}
obj.b();
obj.c();
```

10, window object

10, object this

(iv) What will be the output

```
const Foo = () => { this.name ="Mike";};
const obj = new Foo();
console.log(obj.name);
```

Arrow function dont have constructor So error

Error Foo is not a constructor

(v) What will be the output

```
const Foo = () => {};
Foo.prototype.name = "name"
console.log(Foo.name);
```

Arrow function dont have prototype, So error

(vi) Write code snippet to create arrow function with name profile that takes 2 arguments (name & age) and return object with properties name & age in implicit/implied way (concise body

```
const name ="Mike";
const age = "20";
//Write function here
```

const ar = (name, age) => { return {name, age};}

(vii) Write an arrow function which takes array of integers, and returns the sum of the elements of the array. Use JS reduce method in solution.

## 3. Default Arguments

(i) What will be the output

```
function calc(total, tax=.20, tip=.10){
return total + total*tax + total*tip;
}
const bill = calc(100,null,.2);
console.log(bill);
```

(ii) What will be the output

```
function test(num =1 ){
console.log(typeof num);
}
test("");
```

(iii) Write a function that executes a callback function after a given delay in milliseconds. The default value of delay is one second.

(iv)Change the below code such that the second argument of printComment has a default value that's initially 1, and is incremented by 1 after each call.

```
function printComment( comment , line ) {
console. log( line, comment ) ;
}
```

(v)

Write a function that executes a callback function after a given delay in milliseconds. The default value of delay is one second.

# 4. CURRYING in Js

(i) What will be the output

```
const curriedMultiply = n => m => n * m;
const calc = curriedMultiply(3)(4)
console.log(calc);
```

(ii) Create a function than when executed as follows

greetings('Mike')('Wish you Happy Birthday!')('Steve');

will print in below format (Should maintain line breaks)

Dear Mike,

Wish you Happy Birthday!

From,

Steve

**Note – Use Template literals and Currying technique**