Week 2 Graded Questions

Q1: Which of the following shows the correct output, if the program written below is executed?

let a = [75, 55, 22, 5];
console.log(a.sort());

    A. []
    B. Error
    C. [5, 22, 55, 75]
    D. [22, 5, 55, 75]

Answer: D

Solution: The sort() method sorts the elements in the alphabetical order by default. A user defined function can be passed as an argument to this method in order to customize the default sorting logic.

Q2: Which of the following shows the correct output, if the program written below is executed?

```
a = {
  name: 'Abhi',
  age: 22
};

b = [];
for (let i = 0; i < 3; i++){
    b.push(a)
}

b[1].name = 'Akshay';
console.log(b[0].name);
```

    A. Abhi
    B. Akshay
    C. undefined
    D. ReferenceError

Answer: B

Solution: Here, the same object reference "a" is pushed thrice to an array. Hence, a change in one element of the array will affect all other elements of the array, as all the elements are referencing to the same object.

Q3: Which of the following shows the correct output, if the program written below is executed?

```
const a = {
  name: 'Abhi',
  age: 22
};

b = [];
for (let i = 0; i < 3; i++){
    b.push({...a})
}
b[1].name = 'Akshay';
console.log(b[0].name);
```

A. Abhi
B. Akshay
C. Undefined
D. ReferenceError

Answer: A

Solution: Here, a destructured version of the object reference "a" is pushed thrice to an array. Hence, a change in one element of the array will not impact any other element of this array, as all the elements are referencing to the different objects.

Q4: Consider the following program,

```
x = [1, 2, 3, 4, 5, 6]
y = [...x, 7, 8, 9]
z = y.filter((x) => x % 2 == 0)
    .map((i) => i * i)
    .reduce((a, i) => a + i, (a = 0))
```

```
console.log(z)
```

What will be the output of the program?

    A. 100
    B. 120
    C. 140
    D. 160

Answer: B

Solution: Here, the filter() method returns an array of the even elements from the array "y". The result of which is given to the map() method, which returns an array of square of elements from the array it got from the filter() method as the result. The resultant array is then passed to the reduce() method, which computes the final answer as 120, which further gets displayed on the console.

Q5: What will be the output of the following program?

```
const obj = {
  x: 1,
  y: 1,
  set nums(xCommay) {
    numbers = xCommay.split(',')
    this.x = numbers[0]
    this.y = numbers[1]
  },
  get nums() {
    return this.x + ',' + this.y
  },
  xPowery: function () {
    let result = 1
    for (let i = 1; i <= this.y; i++) {
      result = result * this.x
    }
    return result
  },
}
obj.nums = '2,3'
console.log(obj.xPowery())
```

A. 6
B. 5
C. 8
D. None of the above

Answer: C

Solution: The example uses getter and setter properties of an object. The setter modifies the properties of the object, whereas the getter returns the properties. Then, it invokes a method, which does some computations and returns the result.

Q6: What will be the output of the following program?

```
const ePowerx = {
  x: 1,
  n: 1,
  set parameters(param) {
    param = param.split(' ')
    this.x = param[0]
    this.n = param[1]
  },
  get uptoNterms() {
    let result = 1
    let y = 1
    for (let i = 1; i < this.n; i++) {
      y = y * this.x
      result = result + y
    }
    return result
  },
}
ePowerx.parameters = '2 3'
console.log(ePowerx.uptoNterms)
```

A. 3
B. 5
C. 7
D. 15

Answer: C

Solution Above is the logic to compute $1+x+x^{2}+x^{2}+...$ upto n terms and in this problem x=2 and n=3 so, the answer is 1+2+4=7.

Q7: What will be the output of the following program?

```
const course = {
  courseName: 'Modern Application Development 2',
  courseCode: 'mad2',
}

const student = {
  __proto__: course,
  studentName: 'Rakesh',
  studentCity: 'Delhi',
}

const { courseName } = student
console.log(courseName)
```

A. Undefined
B. Modern Application Development 2
C. Will throw syntax error
D. None of the above

Answer: B

Solution: Student object inherit from the course object so, it will have access to all the properties of course object so, Option B is correct.

Q8: Consider the following,

Index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Document</title>
  </head>
  <body>
```

```
    <div id="div1" style="background-color: red"></div>
  </body>
  <script src="script.js"></script>
</html>
```

script.js:

```
var height = 0
var width = 0
var divStyle = document.getElementById('div1').style
let res = setInterval(() => {
  height += 50
  width += 50
  divStyle.height = height + 'px'
  divStyle.width = width + 'px'
  console.log(height)
}, 1000)

setTimeout(() => {
  clearInterval(res)
}, 10050)
```

Assuming that the index.html and script.js are kept inside the same directory, then what will be the height of the HTML element having ID 'div1' after 20 seconds, if rendered using a browser?

    A. 500px
    B. 1000px
    C. 0px
    D. None of the above

Answer: A

Solution: The callback function inside the setInterval will get executed every 1 second but because after 10050 milliseconds callback function inside the setTimeout will get executed and will clear the interval. And, the callback function inside setInterval increases the height of div by 50px, and it will run 10 times so, its final height will be 500px.

Q9: Consider the following,

Index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <script src="module1.js" type="module"></script>
  </head>
  <body>
    <div id="div1" style="border: 2px solid black"></div>
  </body>
</html>
```

module1.js:

```
import { divStyle1, divStyle2 } from './module2.js'
divStyle1.width = "50px"
const div1Style = document.getElementById('div1').style
div1Style.backgroundColor = divStyle1.color
div1Style.height = divStyle2.height
div1Style.width = divStyle1.width
```

module2.js:

```
export const divStyle1 = {
  color: 'red',
  height: '100px',
  width: '100px',
}

export const divStyle2 = {
  color: 'green',
  height: '50px',
}
```

Assuming that the index.html, module1.js, and module2.js are kept inside the same directory, then what will be the background color, height and width of the HTML element having ID 'div1, if rendered using a browser?

   A. Red, 100px, 100px
   B. Red, 50px, 50px
   C. Green, 50px, 100px
   D. Green, 50px, 50px

Answer: B

Solution: width property of divStyle1 has been override inside the module 1 so, width of the div1 is 50px, and backgroundColor comes from divStyle1 so, it will be red and height comes from divStyle2 so, it will be 50px.

Q10: What will be the output of the following program?

```
api = { A: 'Application', P: 'Programming', I: 'Interface' }
const standsFor = (x) => {
  for (const [k, v] of Object.entries(api)) {
    if (k != x) {
      return v
    }
  }
}

console.log(standsFor('A'))
```

A. Application
B. Programming
C. Interface
D. A

Answer: B

Solution: Program will log the first value whose key is not 'A'. So, it will be "Programming".