Week 2 Practice Questions

Q1: Which of the following programs (code snippet 1 and code snippet 2) will generate an error, if executed?

```
Code snippet 1: for (const i = 0; i <= 5; i++) {
                        console.log(i)
                }
```

```
Code snippet 2: for (const i in [1, 2, 3, 4, 5]) {
                        console.log(i)
                }
```

A. Code snippet 1
B. Code snippet 2
C. Both Code snippets 1 and 2
D. None of the above

Answer: A

Solution: In code snippet 1, the value of a const variable is attempted to be incremented in the same scope, which causes the termination of the program with an error, whereas in code snippet 2, a new variable will be created in a new scope with each iteration, so it will not cause any error, and the program will complete its execution without any errors.

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

```
let a = [2, 3, 4];
let b = [2, 2, ...a, 5];
let c = b.find(x => x % 2);
console.log(c);
```

A. 1
B. 2
C. 3
D. 4

Answer: C

Solution: The find() method returns the first element of an array that satisfies the condition specified.

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

```
let a = {
      'name' : 'abhi',
      'age' : 22,
      'place' : 'delhi'
      };
let { name : alt } = a;
console.log(name);
```

A. abhi
B. alt
C. Reference Error
D. None of the above

Answer: C

Solution: The object reference “a” is destructured, and the “name” attribute is imported and renamed as “alt”. No “name” variable or reference exists in the current scope.

Q4: What will be the output of the following program?

```
arr = ['iitmonline', true, 3, (a, b) => a + b]
let result = Array()
for (let i = 0; i < arr.length; i++) {
  result.push(typeof arr[i])
}
console.log(result)
```

A. ['string', 'boolean', 'number', 'function']
B. ['iitmonline', true, 3, (a, b) => a + b]
C. [object, object, object, object]
D. None of the above

Answer: A

Solution: The first element of the array “arr” is of type “string”, the second element is of type “boolean”, the third element is of type “number”, and the last element is of type “function”.

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

```
arr = ['iitmonline', true, 3, (a, b) => a + b]
let result = Array()
for (const i in arr) {
  result.push(arr[i] * arr[i])
}
console.log(result)
```

A. [NaN, NaN, 9, NaN]
B. [NaN, 1, 9, NaN]
C. [1, 1, 9, NaN]
D. None of the above

Answer: B

Solution: A string literal cannot be multiplied with a string literal, which has at least one alphabetical character, and it returns “NaN”, if tried. Similarly, a function cannot be multiplied with a function. However, a boolean value “true” is translated to literal 1 and “false” to 0, if multiplied with another boolean.

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

```
fruit = {
  name: 'Apple',
  color: 'red',
}

let description = ({ name, color, shape = 'Spherical' }) => {
  console.log(`${name} is ${color} and ${shape}`)
}

description(fruit)
```

A. It will throw syntax error
B. Apple is red and undefined
C. Undefined is undefined and Spherical

D. Apple is red and Spherical

Answer: D

Solution: name and color comes from the fruit object after destructing but shape property is not there in fruits, it will take default value ‘Spherical’.

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

```
class Player {
  constructor(name) {
    this.name = name
    this.team = 'Indian Cricket Team'
    this.nationality = 'Indian'
  }
}


class Bowler extends Player {
  constructor(name, wicket, average) {
    super(name)
    this.role = 'Bowler'
    this.wicket = wicket
    this.average = average
  }
}

bumbum = new Bowler('Jasprit Bumrah', 101, 22.79)
console.log(bumbum)
```

A. Bowler {
```
  name: 'Jasprit Bumrah',
  team: 'Indian Cricket Team',
  nationality: 'Indian',
  role: 'Bowler',
  wicket: 101,
  average: 22.79
}
```

B. Bowler {
```
  name: 'Jasprit Bumrah',
```

  team: undefined,
  nationality: undefined,
  role: 'Bowler',
  wicket: 101,
  average: 22.79
}

C. Bowler {
  name: undefined',
  team: undefined,
  nationality: undefined,
  role: 'Bowler',
  wicket: 101,
  average: 22.79
}

D. None of the above

Answer: A

Solution: Bowler inherits from Player. So, Bowler has access to all the properties and method of Player. So, the answer is A

Q8: What will be the output of the following program?

```
class Player {
  constructor(name, team) {
    this.name = name
    this.team = team
  }

  get describe() {
    return `${this.name} from ${this.team} is a ${this.role}`
  }
}

class Batsman extends Player {
  constructor(name, team) {
    super(name, team)
    this.role = 'Batsman'
  }
}

p = new Batsman('Rohit', 'Indian Cricket Team')
console.log(p.describe)
```

A. undefined from undefined is a Batsman
B. Rohit from Indian Cricket Team is an undefined
C. undefined from Indian Cricket Team is a Batsman
D. Rohit from Indian Cricket Team is a Batsman

Answer: D

Solution: Batsman inherits from Player. So, Bowler has access to all the properties and method of Player. So, the answer is D.

Q9: What will be the output of the following program?

```
const obj = {
  a: 10,
  operation(x, y, n) {
    return x ** n + y + this.a
  },
}

const arr = Array()
p = obj.operation
arr.push(p.bind(obj, 2)(3, 2))
arr.push(p.apply(obj, [2, 2, 3, 4, 5]))
arr.push(p.call(obj, 2, 3, 4))
console.log(arr)
```

A. [17, NaN, 29]
B. [17, 20, 29]
C. [NaN, 20, 29]
D. Syntax error

Answer: B

Solution: p.bind(obj, 2)(3, 2) will pass the value of x = 2, y = 3 and n = 2 so, it will push 17 into arr.

p.apply(obj, [2, 2, 3, 4, 5]) will pass 2, 2, 3 for x, y and n so, it will push 20 in arr.
p.call(obj, 2, 3, 4) will pass 2, 3, 4 for x, y and n respectively so, it will push 29 in the arr.
So, option B is correct.

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

```
const obj = {
  firstName: 'Narendra',
  lastName: 'Mishra',

  get fName() {
    return this.firstName
  },

  get lName() {
    return this.lastName
  },

  set lName(name) {
    this.lastName = name
  },
}

obj.lName = 'Mourya'
obj.lName = console.log(obj.lName)
```

A. Mishra
B. Mourya
C. Syntax Error
D. None of the above

Answer: B

Solution: obj.lname = ‘Mourya’ will trigger the set lname() method and change the lastName property and obj.lname will trigger the get lname() method so, will return lastName property. So, option B is correct.