

**Question - 1**  
**Custom Error**

SCORE: 5 points

Error Handling

Javascript

Easy

Type Checking

Variable Scopes

What is the output of this JavaScript code snippet?

```
class CustomError extends Error {
  constructor(message) {
    super(message);
    this.name = "CustomError";
  }
}

function validateData(data) {
  if (!data || typeof data !== "object") {
    throw new CustomError("Invalid data provided");
  }

  if (!data.hasOwnProperty("id") || typeof data.id !== "number") {
    throw new CustomError("Data object is missing an id property");
  }

  return true;
}

const data1 = { id: 42 };
const data2 = "Invalid data";

try {
  validateData(data1);
  console.log("Data1 is valid");
  validateData(data2);
  console.log("Data2 is valid");
} catch (error) {
  if (error instanceof CustomError) {
    console.log(`CustomError: ${error.message}`);
  } else {
    console.log(`Error: ${error.message}`);
  }
}
```

☒

Data1 is valid

CustomError: Invalid data provided

☐

Data1 is valid

CustomError: Data object is missing an id property

☐

Data1 is valid

Error: Invalid data provided

☐ No output

## Question - 2

SCORE: 5 points

SetTimeout

Functions

Javascript

events

Easy

Loops

Javascript Timers

Arrays

What is the output of this JavaScript code snippet?

```
const numbers = [1, 2, 3, 4, 5];

for (var i = 0; i < numbers.length; i++) {
  setTimeout(() => {
    console.log(`Number: ${numbers[i]}`);
  }, i * 1000);
}
```

- ☐ Number: 1  
Number: 2  
Number: 3  
Number: 4  
Number: 5
- ☒ Number: undefined  
Number: undefined  
Number: undefined  
Number: undefined  
Number: undefined
- ☐ Number: 5  
Number: 5  
Number: 5  
Number: 5  
Number: 5
- ☐ No output will be displayed.

## Question - 3

SCORE: 5 points

Closures

Javascript

Javascript Events

Closures

Easy

Closure

What is the output of the following code?

```
const series = (...funcs) => (arg) => {
  return funcs.reduce((prev, curr) => curr(prev), arg);
};
```

```
const result = series(num => num * 10, num => num + 20, num => num / 2, num => num - 5)(4);
```

- ☐ 20
- ☒ 25
- ☐ 30
- ☐ 0

#### Question - 4

##### JavaScript: Objects

SCORE: 5 points

JavaScript Easy Error Handling

Three of the following snippets will create a new JavaScript object with the name "Alex" and age 25. Which one uses incorrect syntax?

☐

```
let person = new Object({name: "Alex", age:25});
```

☐

```
let person = {name: "Alex", age: 25};
```

☐

```
let person = {}  
person.name = "Alex";  
person.age = 25;
```

☒

```
let person;  
person['name'] = "Alex";  
person['age'] = 25;
```

#### Question - 5

##### JavaScript: Typcasting Functions I

SCORE: 5 points

JavaScript Easy Hoisting

Match the JavaScript typcasting functions with their respective output.

Function	Output
1.String([])	A. 0

2. <i>String({})</i>	B. -Infinity
3. <i>Number(null)</i>	C. "[object Object]"
4. <i>Number("-Infinity")</i>	D. ""

- ☐ 1. A  
2. B  
3. D  
4. C
- ☒ 1. D  
2. C  
3. A  
4. B
- ☐ 1. A  
2. D  
3. B  
4. C
- ☐ None of the above

Question - 6

JavaScript: Typecasting Functions II

SCORE: 5 points

JavaScript

Easy

Match the JavaScript typecasting functions with their respective output.

Function Call	Output
1. <i>Boolean(Infinity)</i>	a. 20
2. <i>Number([20, 30])</i>	b. true
3. <i>Boolean("")</i>	c. NaN
4. <i>Number(["20"])</i>	d. false

- ☒ 1-b  
2-c  
3-d  
4-a
- ☐ 1-d  
2-a  
3-b  
4-c
- ☐ 1-d  
2-b  
3-a  
4-c

☐ None of the above

### Question - 7

SCORE: 5 points

#### Declare variables with const keyword in JavaScript

Easy

Javascript Aptitude

One way to declare a variable in JavaScript is to use the keyword *const*. Which of the following are true about *const*? Choose one or more.

- ☐ A variable declared with const is always globally scoped.
- ☒ A variable declared with const is always block-scoped.
- ☐ A variable declared with const can be re-declared.
- ☐ A variable declared with const can be assigned to later.

### Question - 8

SCORE: 5 points

#### Declare variables with let keyword in JavaScript

Easy

Javascript Aptitude

One way to declare a variable in JavaScript is to use the keyword *let*. Which of the following are true about *let*? Choose one or more.

- ☐ A variable declared with let is always globally scoped.
- ☒ A variable declared with let is always block-scoped.
- ☐ A variable declared with let can be re-declared.
- ☒ A variable declared with let can be assigned to later.

### Question - 9

SCORE: 5 points

#### JSON data format

Easy

Javascript Aptitude

JSON uses a few strict rules to encode its content. Which of the following are true about those rules? Choose one or more.

- ☒ JSON uses double quotation marks to delimit strings.
- ☐ JSON uses single quotation marks to delimit strings.
- ☒ Trailing commas are forbidden.
- ☐ Trailing commas are allowed and recommended because of cleaner diffs when working with version control systems.

### Question - 10

SCORE: 5 points

#### localStorage object

The localStorage object is used to store the data locally. Which of the following are true about localStorage objects? Select one or more.

- ☒ The data in localStorage is stored without an expiration date.
- ☐ localStorage allows for several data types, including numbers and strings.
- ☒ All data values stored in localStorage are stored as strings.
- ☐ localStorage is used to store limited data sizes. It is usually restricted to just a few megabytes.

## Question - 11

SCORE: 5 points

### Browser Display

JavaScript

Document Object Model

Functions

Easy

Using JavaScript, check whether the browser is online. Display "*I am online*" if it is. Otherwise, display "*I am offline*".  
If the browser is online, display the current *latitude* and *longitude* of your location.

```
<p id="status"></p>
<p id="location"></p>
```

The `<p>` with `id="status"` is used to display the status: "*I am offline*" or "*I am online*".

The `<p>` with `id="location"` is used to display the latitude and longitude in the format below:

Latitude: 12.9321589

Longitude: 77.61325529999999

- ☒

```
function checkStatus() {

    if(navigator.onLine){
        document.getElementById("status").innerHTML = "I am online";
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(printLocation);
        }

    }
    else

        document.getElementById("status").innerHTML = "I am offline";
}
function printLocation(position) {
    document.getElementById("location").innerHTML = "Latitude: " + position.coords.latitude +
    "<br/>Longitude: " + position.coords.longitude;
}
}
```
- ☐

```
function checkStatus() {

    if(navigator.onLine){
```

```

document.getElementById("status").innerHTML = "I am online";
if (navigator.geolocation) {
navigator.geolocation.getCurrentPosition(printLocation);
}

}
else

document.getElementById("status").innerHTML = "I am offline";
}
function printLocation(position) {
document.getElementById("location").innerHTML = "Latitude: " + position.coords.latitude +
"<br/>Longitude: " + position.coords.longitude;
}
}

☐ function checkStatus() {

    if(navigator.online){
        document.getElementById("status").innerHTML = "I am online";
        if (navigator.geolocation) {
navigator.geolocation.getCurrentPosition(printLocation);
        }

    }
    else

        document.getElementById("status").innerHTML = "I am offline";
}
function printLocation(position) {
document.getElementById("location").innerHTML = "Latitude: " + position.coords.latitude +
"<br/>Longitude: " + position.coords.longitude;
}
}

☐ function checkStatus() {

    if(navigator.onLine){
        document.getElementById("status").innerHTML = "I am online";
        if (navigator.geolocation) {
navigator.geolocation.getCurrentPosition(printLocation);
        }

    }
    else
        document.getElementById("status").innerHTML = "I am offline";
}
function printLocation(position) {
document.getElementById("location").innerHTML = "Latitude: " + position.coords.latitude +
"<br/>Longitude: " + position.coords.longitude;
}
}

```

## Question - 12

### Code Prediction

SCORE: 5 points

JavaScript

Easy

Loops

Variable Scopes

Predict the output of the following code:

```
function Education() {
    this.education = "B.Tech";
}

function main() {
    Person = new Education();
    Person.name = "Michael";
    Person.surname = "Hicks";
    var Person;
    Person.residence = "Bangalore";
    Person.age = 23;
    Object.freeze(Person);
    console.log(delete Person.name);

    for (const item in Person) {
        console.log(item);
    }
}
```

- ☐ False education name surname residence age
- ☒ false education name surname residence age
- ☐ True education surname residence age
- ☐ true education surname residence age
- ☐ false education surname residence age
- ☐ true education name surname residence age
- ☐ Compile time error

### Question - 13

#### Array Element Passing

SCORE: 5 points

JavaScript   Exception Handling   Arrays   Closure   Loops   Easy

Examine the code snippet below. You have to identify the correct snippet for the main and pass each element of the array *list=[0,4,1,3]* such that for every number less than 2, the error message, *"num should be greater than 2"*, is displayed.

Otherwise, the value returned from the function *twice* is displayed. What could be the possible solution(s)?



```
function Myfunc() {
  this.twice = function(num) {
    if (num < 2) {
      throw "num should be greater than 2"
    }
    return num * 2;
  }
}
```

☒ `const myfunc = new Myfunc();`  
`const list = new Array(0, 4, 1, 3);`  
`for (let i = 0; i < 4; i++) {`  
 `try {`  
 `const res = myfunc.twice(list[i]);`  
 `console.log(res);`  
 `} catch (e) {`  
 `console.log(e);`  
 `}`  
`}`

☐ `const myfunc = new Myfunc();`  
`const list = {0, 4, 1, 3};`  
`for (let i = 0; i < 4; i++) {`  
 `try {`  
 `const res = myfunc.twice(list[i]);`  
 `console.log(res);`  
 `} catch (e) {`  
 `console.log(e);`  
 `}`  
`}`

☒ `const myfunc = new Myfunc();`  
`const list = [0, 4, 1, 3];`  
`for (let i = 0; i < 4; i++) {`  
 `try {`  
 `const res = myfunc.twice(list[i]);`  
 `console.log(res);`  
 `} catch (e) {`  
 `console.log(e);`  
 `}`  
`}`

☐ `const myfunc = new Myfunc();`  
`const list = [0, 4, 1, 3];`  
`try {`  
 `for (let i = 0; i < 4; i++) {`  
 `const res = myfunc.twice(list[i]);`  
 `console.log(res);`  
 `}`  
`} catch (e) {`  
 `console.log(e);`  
`}`

## Question - 14

### First Five Alerts

SCORE: 5 points

Closures

JavaScript

Easy

Closure

Select the option that has the correct sequence of first five alert messages, when the following javascript code snippet is run:

```
function func(x) {
  var z = 8;

  return function(y) {
```

```
alert(x + y + z);
}
}

var n1 = new Number(32);
var a1 = func(n1);
var n2 = new Number(16);
var a2 = func(n2);
var n3 = new Number(8);
var a3 = func(n3);
var n4 = new Number(4);
var a4 = func(n4);
var n5 = new Number(2);
var a5 = func(n5);
a1(2);
a2(4);
a3(8);
a4(16);
a5(32);
```

- ☐ 34, 36, 40, 48, 64
- ☐ 2, 4, 8, 16, 32
- ☐ 24, 24, 28, 28, 42
- ☒ 42, 28, 24, 28, 42

### Question - 15

SCORE: 5 points

Which of the following is not true of Javascript?

JavaScript Web Development Easy Javascript Aptitude

Which of the following is not true of Javascript?

- ☐ Javascript is case sensitive
- ☒ Semicolons at the end of Javascript statements are required
- ☐ Javascript statements can be grouped together in blocks
- ☐ Javascript is loosely typed

### Question - 16

SCORE: 5 points

What is the correct way to create a javascript array?

JavaScript Web Development Easy Javascript Aptitude Arrays

What is the correct way to create a javascript array?

- ☒ var items = ["Orange", "Apple"];

- ☐ var items = {"Orange","Apple"};
- ☐ var items = new array("Orange","Apple");
- ☐ var items[] = {"Orange","Apple"};

### Question - 17

SCORE: 5 points

What is the data type of *null* in Javascript?

JavaScript Web Development Easy Javascript Aptitude

What is the data type of *null* in Javascript?

- ☐ undefined
- ☐ integer
- ☒ object
- ☐ none

### Question - 18

SCORE: 5 points

Which of the following javascript functions perform the given tasks correctly?

JavaScript Web Development Easy ES6 Features

Which of the following javascript functions perform the given tasks correctly?

- The function takes one numeric variable as parameter.
- If the variable is 0 then it creates a dialogue box which says "zero".
- If the variable is 1 then it creates a dialogue box which says "one".
- Otherwise it creates a dialogue box which says "Invalid input".

☐

```
const myFunction = value => {  
  switch (value) {  
    case 0:  
      str = "0";  
      break;  
    case 1:  
      str = "1";  
      break;  
    default:  
      str = "Invalid input";  
      break;  
  }  
  return alert(str);  
}
```

☐

```
const myFunction = value => {  
  switch (value) {  
    case 0:  
      str = "zero";  
    case 1:  
      str = "one";  
    default:
```

- ```
    str = "Invalid input";
  }
  alert(str);
}
```
- ☐ const myFunction = value => {  
 switch (value) {  
 case "0":  
 str = "zero";  
 break;  
 case "1":  
 str = "one";  
 break;  
 default:  
 str = "Invalid input";  
 break;  
 }  
 alert(str);  
}
- ☒ const myFunction = value => {  
 switch (value) {  
 case 0:  
 str = "zero";  
 break;  
 case 1:  
 str = "one";  
 break;  
 default:  
 str = "Invalid input";  
 break;  
 }  
 alert(str);  
}

### Question - 19

SCORE: 5 points

#### JS Data Types

JavaScript

Web Development

Easy

Javascript Aptitude

Which of the following is not a primitive Datatype in Javascript?

- ☒ Array
- ☐ Number
- ☐ String
- ☐ Boolean

### Question - 20

SCORE: 5 points

#### Loops in JavaScript

JavaScript

Web Development

Easy

Loops

Which of the following is not a looping structure in JavaScript?

- ☐ for
- ☐ while
- ☐ do - while
- ☒ do - if
- ☐ for each

**Question - 21**  
Which of the following is not a core datatype?

SCORE: 5 points

Python Easy

Which of the following is not a core datatype?

- ☐ Lists
- ☐ Dictionary
- ☐ Tuples
- ☒ Class

**Question - 22**  
What is the output?

SCORE: 5 points

JavaScript Web Development Easy Type Checking

What is the value of y after the following code?

```
let x;  
let y = x === null;
```

- ☒ false
- ☐ true
- ☐ undefined
- ☐ null

**Question - 23**  
How can you add a comment in a JavaScript?

SCORE: 5 points

Web Development JavaScript Front-End Development Easy

How can you add a comment in a JavaScript?

- ☒ //This is a comment
- ☐ REM This is a comment-->
- ☐ 'This is a comment
- ☐ #This is a comment

### Question - 24

#### JavaScript Assignment

SCORE: 5 points

Language Proficiency

JavaScript

Problem Solving

Easy

Javascript Aptitude

Consider the following JavaScript code snippet:

```
function foo () {  
    return 5  
}
```

What will the code `let myVar = foo;` do?

- ☐ Assign the integer value *5* to the variable *myVar*
- ☒ Assign a reference to the *foo* function to the variable *myVar*
- ☐ Throw an exception
- ☐ Nothing