Server side JavaScript with Node.js

Week 1
85% marks

Question 1
What is the use of the Underscore Variable in REPL session?
1 point

To get the last command used.

To get the last result.

To store the result.

None of the above.

2.
Question 2
Node.js is a _____language.
1 point

server side

client side

middleware

None of the above.

3.
Question 3
Node.js = _____+ _____.
1 point

Compiler + The Javascript Library

Runtime Environment + The JavaScript Library

Interpreter + The JavaScript Library

JavaScript Library

4.
Question 4
The v8 engine works inside the_____of the browser.
1 point

Network API

Core OS

DOM context

Server side JavaScript with Node.js

File system

5.

Question 5
All APIs of Node.js libraries are _____.
1 point

synchronous

blocking

unblocking

asynchronous

6.

Question 6
_____is the package manager for Node.
1 point

npn

mpm

npm

None of the above.

7.

Question 7
_____is a Node.js component.
1 point

DOM

Node CLI

package-lock.json

None of the above.

Question 8
Asynchronous Jobs run on_____threads.
1 point

worker

executable

daemon

Multiple

9.

Question 9

Node.js is not suited for_____since it is single threaded.

1 point

IO intensive operations

File Intensive Operations

CPU intensive operations

None of the above.

10.

Question 10

Variables are_____for storing data.

1 point

containers

compilers

integrators

Controllers

11.

Question 11

The const declaration creates a_____reference to a value.

1 point

read-write

read-only

write-only

None of the above.

12.

Question 12

What is the output of the code snippet given below? console.log(age)

```
var age = 30

console.log(age)
```

1 point

undefined

30

undefined 30

error in code

13.

Question 13
The process object is a_____object.
1 point

global

local

hoisted

None of the above.

14.

Question 14
Command line arguments can be accessed through the_____functionality.
1 point

process

process.arg

process.argc

process.argv

15.

Question 15
The "function" and " var" are known as _____.
1 point

keywords

datatypes

declaration keywords

prototypes

16.

Question 16
In the following syntax of the switch statement, the Expression is compared with the labels using which of the following operators? switch(expression) { statements }
1 point

"==="

"=="

"="

equals

17.

Question 17
What is the output of the code snippet given below? var count =0; while (count ```<10) { console.log(count); count++; }
1 point

Infinite loop

Prints values from 1 to 10

Prints values from 0 to 9

Prints undefined

18.

Question 18
What is the output of the code snippet given below? var stringValue = "40"; var intValue = 50; console.log(stringValue + intValue);
1 point

90

40

error

4050

19.

Question 19
What is the output of the code snippet given below? var x = 0 while (x != 0) { if(x == 1) continue; else x++; }

console.log(x)
1 point

0

infinite loop

1

None of the above.

20.

Question 20
What is the output of the code snippet given below? var a=0; var b =0; while (a ```<3) { a++; b += a; console.log(b); }
1 point

1,1,1

1,3,6

1,3,7

1,3,5

Week 2
95% Marks

1.
Question 1

A function with no return value is called_____function.
1 point

Procedure

Method

Static function

Dynamic function

2.
Question 2

What will be the output of a return statement if it does not have an associated expression?
1 point

It returns the value 0.

It will throw an exception.

It returns the undefined value.

None of the above

3.
Question 3

When can we describe a function as optional in JavaScript?
1 point

When the function is defined as a looping statement

When function is defined as expressions

When function is predefined

All of the above

4.

Question 4

Do all the JavaScript functions return a value?

1 point

It is manditory

Not necessary

Few functions return value by default

All of the above

5.

Question 5

What will be the output of the following code snippet? function ab(){ console.log("inside "); } console.log(typeof ab);

1 point

Function

Object

Gives function name

None of the above

6.

Question 6

What will be the output of the given code snippet? var square = function ab(x) { x++; return x * x ; }; console.log (ab (5)); console.log (square (12));

1 point

25 169

36 169

undefined 72

undefined 169

7.

Server side JavaScript with Node.js

Question 7

What will be the output of the given code snippet? console.log (square (6)); var square = function ab(x) { return x * x ; };

1 point

36

square (6)

square is not a function

None of the above

8.

Question 8

What will be the output of the following code snippet? ab (12) ; function ab( ) { return x * x ; }

1 point

144

ab is not defined

ab is not a function

x is not defined

9.

Question 9

Whatwill be the output of function if printed? var make Noise = function () { console.log (" Pling !") ; }; make Noise();

1 point

Pling !

make Noise is not a function

variable name should not contain space : error line 1

None of the above

10.

Question 10

What will be the output of the following code snippet? function(){ console.log("inside"); } function();

1 point

inside

Error at line 1

function declaration contains semicolon at end

None of the above

11.

Question 11

What will be the output of the following code snippet? var f = function (x){ console.log("inside function " + x); }; f(12); console.log(x);

1 point

inside function 12 12

inside function 12 x is not defined

inside function x 12

None of the above

12.

Question 12

What will be the output of the following code snippet? var x = function (a ,b) { var result = 1; for (var count = 0; count ```< b ; count ++) result *= a; return result; }; console.log (x (2 , 10)) ;

1 point

1024

100

20

None of the above

13.

Question 13

Predict the output of the following program: var carMakes = ; console.log('Old array : ' +carMakes.join()); carMakes.splice(2,1, 'ALPHA-ROMEO'); console.log('New array : ' +carMakes.join());

1 point

None of the below

Old array : BMW,AUDI,TOYOTA,SUZUKI New array : BMW,AUDI,TOYOTA,ALPHA-ROMEO,SUZUKI

Old array : New array :

Old array : BMW,AUDI,TOYOTA,SUZUKI New array : BMW,AUDI,ALPHA-ROMEO,SUZUKI

14.

Question 14

Predict the output of the following program: var carMakes = ; console.log('Old array : ' +carMakes.join()); carMakes.splice(2,1); console.log('New array : ' +carMakes.join());

1 point

Old array : BMW,AUDI,TOYOTA,SUZUKI New array : BMW,AUDI,TOYOTA,SUZUKI

Old array : New array :

Old array : BMW,AUDI,TOYOTA,SUZUKI New array : BMW,AUDI,SUZUKI

None of the above

15.

Question 15

Predict the output of the following program: var carMakes = ; console.log('Old array : ' +carMakes.join()); carMakes.sort(); console.log('Sorted array : ' +carMakes.join());

1 point

Old array : Sorted array :

Old array : BMW,AUDI,TOYOTA,SUZUKI Sorted array : AUDI,BMW,SUZUKI,TOYOTA

Old array : BMW,AUDI,TOYOTA,SUZUKI Sorted array : BMW,AUDI,SUZUKI,TOYOTA

None of the above

16.

Question 16

Predict the output of the following program: var carMakes = ; console.log('Old array : ' +carMakes.join()); carMakes.sort(); carMakes.reverse(); console.log('Array in reverse order: ' +carMakes.join());

1 point

Old array : Sorted array :

Old array : Array in reverse order:

Old array : BMW,AUDI,TOYOTA,SUZUKI Array in reverse order: TOYOTA,SUZUKI,BMW,AUDI

None of the above

17.

Question 17

What will be the output of the following code snippet? var a1 = ; var a2 = new Array(3); 0 in a1 ; 0 in a2;

1 point

true false

false true

true true

false false

18.

Question 18

Which of the following statements defines the pop() method?

1 point

Decrements the total length by 1

Increments the total length by 1

Prints the first element but no effect on the length

None of the above

19.

Question 19

What happens if the reverse() and the join() methods are used simultaneously ?

1 point

Reverses and stores in the same array

Reverses and concatenates the elements of the array

Reverses

All of the above

20.

Question 20

Predict the output of the following program: var a = ; a.slice(0,3);

1 point

Returns [1,2,3]

Returns [4,5]

Returns [1,2,3,4]

Returns [1,2,3,4,5]

21.

Question 21

Predict the final output of the following program: var a = ; a.unshift(1); a.unshift(22); a.shift(); a.unshift(3,); a.shift(); a.shift(); a.shift();

1 point

1

[4,5]

[3,4,5]

Execption is thrown

22.

Question 22

What is the use of array map() function?

1 point

Maps the elements of another array into itself.

Passes each element of the array and returns the necessary mapped elements.

Passes each element of the array on which it is invoked to the function you specify, and returns an array containing the values returned by that function.

None of the above

Server side JavaScript with Node.js

Week 3

95% Marks

1.
Question 1

Which of the following statements is true for package.json?
1 point

package.json updates dependencies of Node Application.

package.json is used to define the properties of the package.

package.json is present in the root directory of any Node Application.

All of the above

2.
Question 2

Which of the following modules is required for network specific operations?
1 point

os module

net module

fs module

path module

3.
Question 3

Which of the following commands will show all the locally installed modules?
1 point

npm ls -g

node ls -g

npm ls

node ls

4.

Server side JavaScript with Node.js

Question 4

Which of the following modules is required from Node.js to perform path operations?

1 point

os module

path module

fs module

HTTP module

5.

Question 5

Which of the following options is an incorrect expression to expose a function in Node.js

1 point

module.exports = function calculate(operation, lhs, rhs) {}

exports = function calculate(operation, lhs, rhs) {}

module.exports = exports = function calculate(operation, lhs, rhs) {}

export function calculate(operation, lhs, rhs) {}

6.

Question 6

Which of the following statements imports foo alone in the correct way?

1 point

const foo = require ('./example.js');

const foo = require ('./example');

const { foo } = require ('./example.js');

const { null, foo } = require ('./example.js');

7.

Question 7

Which of the following statements is correct about modules?

1 point

You can have multiple methods and variables exported from a module.

Once you have exported a method, it must refer to valid JavaScript expression.

If you don't export any thing from the module, it will not be usable by other part of your code/project.

All of the above

8.

Question 8

Which of the following classes is used to create the events and also consume them in Node.js?

1 point

EventEmitter

Events

NodeEvent

None of the above

9.

Question 9

What does npm stand for?

1 point

Node project manager

Node Package Manager

New package Manager

New project manager

10.

Question 10

In Node.js, third party module can be updated, deleted, or installed using .

1 point

Node.exe

module.exports

Server side JavaScript with Node.js

Node Package Manager

REPL

11.

Question 11

Single or multiple files organized in JavaScript having simple or complex functionality that can be reused throughout Node.js application are called_____.

1 point

Function

Package

Module

Library

12.

Question 12

Which of the following statements is true for CommonJS modules?

1 point

CommonJS modules are loaded synchronously and processed in the order the JavaScript runtime finds them.

CommonJS module is used only with server side JavaScript.

The CommonJS module specification is the standard used in NodeJS for working with modules.

All of the above

13.

Question 13

Which of the following statements is true for nodemon Module?

1 point

The nodemon Module is a module that develops Node. js based applications by automatically restarting the node application.

It is a logging Module in Node.js

It is an error handling module.

Nodemon has inbuilt methods that help filter data in array and objects.

14.

Question 14

Which of the following code snippets will print the hostname ?

1 point

os.platform()'

os.cpus()'

os.hostname()

os.getHostname()'

15.

Question 15

How will you import any module in Node.js?

1 point

using require() function

using include() function

using module.export function

using module.import function

16.

Question 16

Which of the following options are not in-built modules of Node.js?

1 point

http

fs

stream

Lodash

17.

Question 17

Lodash module is used for .

1 point

Lodash contains tools to simplify programming with strings, numbers, arrays, functions and objects.

It is a JSON logging library for Node.js services.

This module enables interacting with the file system.

This module provides methods to raise and handle events.

18.

Question 18

Which of the following statements is true for Path Module?

1 point

path.dirname() - Returns the directory part of a path

path.isAbsolute() - Returns true if it's an relative path

path.parseInt() - Parses a path to an object with the segments that compose it

path.extname() - Returns the absolute path of a file and directory

19.

Question 19

Which of the following statements is true for node_modules?

1 point

The goal of node_modules file is to keep track of the exact version of every package that is installed and also the location from where they are installed.

This acts as a cache for the external modules that the project depends upon. When npm install is done, the packages get downloaded from the npm registry and are copied into the node_modules folder and Node.js looks for them when you import them.

It is a JSON file that lives in the root directory of your project.

It's the package.json file that enables npm to start the project, run scripts, install dependencies, and publish to the npm registry.

20.

Question 20

Which of the following statements is true for url module?

1 point

Provides information and control about the current Node.js process.

Includes methods to deal with file paths.

Provides utilities for URL resolution and parsing.

Used to handle file system.

Week 4

90% Marks

1.

Question 1

Each function of a JavaScript program will be pushed onto the _____in the order of calling.

1 point

call stack

heap memory

task queue

event loop

2.

Question 2

JavaScript is _____by default.

1 point

asynchronous

synchronous

non-blocking

None of the above

3.

Question 3

Server side JavaScript with Node.js

What are Control Structures for Asynchronous Programming?

1 point

blocks, functions, control statements

functions, keywords, callbacks

callbacks, promises, async await

None of the above

4.

Question 4

A call back method always takes _____ as the first parameter.

1 point

function

error

variable

higher order function

5.

Question 5

A callback function is a function passed into another function as a _____.

1 point

argument

variable

function

none of the above

6.

Question 6

NodeJS retrieves any incoming request and adds them to the _____.

1 point

Callback queue

Event Loop

Event Queue

Thread Pool

7.

Question 7

The Event Loop processes _____requests.

1 point

Blocking

Promise

Callback

Non-blocking

8.

Question 8

Each_____of the event loop maintains a separate callback queue.

1 point

step

phase

loop

none of the above

9.

Question 9

_____is a timer callback.

1 point

setTimeout()

setImmediate()

socket.on()

process.nextTick()

10.

Question 10

The _____queue is for resolving promises.

1 point

process.nextTick()

poll

microtasks

timer callback

11.

Question 11

Each time the event loop takes a full trip completing all the phases, it is called a _____.

1 point

nextTick

tick.

phase

poll

12.

Question 12

A Promise is said to be in pending state when _____.

1 point

the asynchronous operation is not yet complete

the operation successfully completes

when the operation terminates with an error

none of the above

13.

Question 13

_____is a callback that will eventually receive the fulfillment value of the Promise.

1 point

reslove

onRejected

onFulfilled

reject

14.

Question 14

What will be the output of the following code snippet ?

```
const add = new Promise((resolve,reject)=>{
    setTimeout(() => {
       resolve([6,7,8])
       reject('error in code')
     }, 2000);
})

add.then((result)=>{
    console.log("Success !  "+result)
}).catch((error)=>{
console.log(error)
})
```

1 point

Success ! 6,7,8

error - both resolve and reject in same block

error in code

none of the above

15.

Question 15

What will be the output of the following code snippet ?

```
startTime = ()=> {
    const today = new Date()
    let h = today.getHours();
    let m = today.getMinutes();
    let s = today.getSeconds();
    m = checkTime(m);
    s = checkTime(s);
   console.log(h + ":" + m + ":" + s)
    setTimeout(startTime, 1000);
  }


  checkTime = (i) => {
    if (i < 10) {i = "0" + i};    < 10
    return i;
  }
  StartTime()
```

1 point

Prints the current time in hh:mm:ss continuously after every 1 second

Prints the current time in hh:mm:ss format once

Infinite loop

Prints undefined continuously after every one second

16.

Question 16

An async function returns a_____.

1 point

value

function

callback

promise

17.

Question 17

_____helps you define a list of promises, and execute something when they are all resolved.

1 point

Promise.any()

Promise.race()

Promise.all()

Promise.new()

18.

Question 18

What is the output of the below code ?

```
setTimeout(() => {
  console.log('after ')
}, 0)

console.log(' before ')
```

1 point

before, after

after,before

before

after

19.

Question 19

_____use promises behind the scenes.

1 point

await

async functions

callback functions

functions

20.

Question 20

Debugging_____is hard because the debugger will not step over asynchronous code.

1 point

await

async functions

callback functions

promises

Week 5

95% Marks

1.

Question 1

Which of the following statements is true for EventEmitter.emit property?

1 point

emit property is used to locate an event handler

emit property is used to fire an event

emit property is used to bind a function with the event

emit property is used when fileRead happens

Server side JavaScript with Node.js

2.

Question 2

Which of the following statements is false about Streams?

1 point

Handles back pressure

Can pause and resume stream operation

Streams can be on Object mode

Only Asynchronous operations can be performed

3.

Question 3

Which of the following methods of fs module is used to get information about a file?

1 point

fs.open(path, flags , callback)

fs.readFile(path, flags , callback)

fs.stat(path, callback)

fs.watchFile(path, callback)

4.

Question 4

Which of the following methods of fs module is used to read a directory?

1 point

fs.readDirectory(path , callback)

fs.read(path , callback)

fs.readdir(path, callback)

None of the above

5.

Question 5

Which of the following statements is true for File I/O operations in Node application?

1 point

Server side JavaScript with Node.js

NodeJS implements File I/O using simple wrappers around standard POSIX function.

To work with File I/O fs module needs to be imported

All the File I/O operations(read , write, append) are asynchronous by default

6.

Question 6

Which of the following statements is true for EventEmitter.on property?

1 point

on property is used to locate an event handler

on property is used to bind an event with a function

on property is used to bind a function with the event

on property is used to fire an event

7.

Question 7

Which of the following fs module methods is used to close the file?

1 point

fs.close(fd, callback)

fs.closeFile(fd, callback)

fs.closePath(fd, callback)

fs.closefile(fd, callback)

8.

Question 8

Which of the following events is not supported by Readable Streams in NodeJS?

1 point

Event data

Event end

Event error

Event cork

9.

Question 9

Which of the following is a benefit of using Stream processing?

1 point

Low memory footprint by the application

Consistent way for Asynch & Synch processing

Faster processing of the data

All of the above

10.

Question 10

Which of the following API methods is not supported for EventEmitter?

1 point

emitter.observe

emitter.once

emitter.emit

emitter.on

11.

Question 11

Which of the following Classes is used to implement NodeJS Streams?

1 point

Memory Buffers

Event Loop

Promises

EventEmitters

12.

Server side JavaScript with Node.js

Question 12

Which of the following types of stream is not supported in NodeJS?

1 point

Readable Stream

Writable Stream

Transform Stream

None of the above

13.

Question 13

Which of the following statements is false for Buffer class?

1 point

It represents a fixed-size chunk of memory (can't be resized).

It is implemented by the NodeJS Buffer class.

The Buffer object is a global object in NodeJS, and it is not necessary to import it using the require keyword.

To use the Buffer object we need to import the global Buffer Object by writing require('Buffer')

14.

Question 14

Which of the following scenarios is possible using Streams?

1 point

Read from file as stream and pipe to another file

Read incoming API request as stream and return response as Stream

Read data from Databases as stream

All of the above

15.

Question 15

Which of the following events is not supported by Writable Streams in NodeJS?

1 point

Event data

Event `drain

Event pipe

Event unpipe

16.

Question 16

Which of the following classes is used to create custom event in NodeJS?

1 point

Event

EventEmitter

Buffer

All of the above

17.

Question 17

Which of the following scenarios makes the best or ideal case for using NodeJS?

1 point

I/O Intensive operations

Concurrent data requests

Data stream processing applications

All of the above

18.

Question 18

Which of the the following methods appends specified content to a file?

1 point

fs.appendFile()

fs.open()

fs.writeFile()

None of the above

19.

Question 19

Which of the following modules is used to implement custom stream?

1 point

require('fs')

require('http')

require('stream')

require('events')

20.

Question 20

Which of the folloiwng methods can be used to read a file asynchronously?

1 point

fs.readFileSync(path, options)

fs.readFile(filename, encoding,callback)

fs.read(filename)

None of the above

Week 6
100% Marks

1.

Question 1

What are the types of errors that can occur in a Node.js application?

1 point

Operational and Logical Errors

Syntax and Semantic errors

Compile time and Runtime errors

None of the above

2.

Question 2

System out of memory is a_____error

1 point

Logical

Operational

Syntax

Runtime

3.

Question 3

The JS environment does not detect a _____error

1 point

Semantic

Operational

Logical

None of the above

4.

Question 4

ReferenceError is a _____error

1 point

Assertion error

User-defined error

System error

Standard JS Error

5.

Question 5

Errors in Node.js are handled through _____

1 point

objects

Error classes

Exceptions

JS libraries

6.

Question 6

To throw an Error object explicitly we use the_____keyword

1 point

throws

throw

try …. Catch

finally

7.

Question 7

The try block contains the _____code that can throw an error

1 point

critical

normal

control flow

None of the above

8.

Question 8

Server side JavaScript with Node.js

_____is not a constructor of the Error class

1 point

new Error()

new Error(message)

new Error(filename)

new Error(message, options)

9.

Question 9

Synchronous APIs will use _____to report errors implicitly

1 point

throw

throws

new

try..catch

10.

Question 10

An async functions can have _____blocks

1 point

throws

try..catch

finally

function

11.

Question 11

What command is used to run the inbuilt debugger ?

1 point

node <name of the .js file><parameters>

node start <name of the .js file><parameters>

node inspect <name of the .js file><parameters>

node debug <name of the .js file><parameters>

12.

Question 12

The _____statement is attached in the program to invoke the inbuilt debugger through running the inspect command.

1 point

debug

debugger

start debugger

begin debug

13.

Question 13

_____is a place in the program where the execution is stopped by the debugger

1 point

Step over

Step into

Breakpoint

None of the above

14.

Question 14

_____window is used to observe more than one variable

1 point

Watch

Call stack

debugger

explorer

15.

Question 15

_____serves as a means to monitor, observe and optimize software development

1 point

Software Debugging

Software Diagnosis

Software Testing

None of the above

16.

Question 16

_____is a tool that is built into the Node.js core

1 point

Diagnosis Monitor

Report

Diagnostic Report

None of the above

17.

Question 17

The diagnostics report can be written to a _____file

1 point

.csv

.txt

.js

.json

18.

Question 18

The _____object helps to generate the diagnosis report.

1 point

process

prototype

local

None of the above

19.

Question 19

_____triggers diagnostic reporting on fatal errors when true

1 point

reportOnSignal

reportOnFatalError

reportOnUncaughtException

reportOnException

20.

Question 20

The _____terminal is used to execute code in the debug mode in Node.js

1 point

JavaScript Debug

powershell

command prompt

debugger

Server side JavaScript with Node.js

Week 7

86% Marks

1.

Question 1

Predict the output of the following code snippet:

```
function makeAdder(a) {
    return function(b) {
        return a + b;
    };
}
var add5 = makeAdder(5);
add5(6);
```

1 point

6

5

11

error

2.

Question 2

Predict the output of the following code snippet:

```
let str = 'Selenium WebDriver';
console.log(str.includes('Web', 10));
```

1 point

0

1

3.

Question 3

Server side JavaScript with Node.js

Predict the output of the following code snippet:

```
function subtract( x = y, y = 1 ) {
    return x - y;
}
subtract(10);
```

1 point

9

10

1

error

4.

Question 4

Predict the output of the following code snippet:

```
var automationtools = ["protractor", "cypress", "selenium", "cucumber"];  auto
mationtools.splice(1, 1, "watir", "uft");
```

1 point

protractor,watir,selenium,cucumber

protractor,watir,uft,selenium,cucumber

watir,uft,cypress,selenium,cucumber

watir,cypress,selenium,cucumber

5.

Question 5

Predict the output of the following code snippet: var iyal =; iyal ='puram'; console.log (iyal.length)

1 point

4

100

101

None of the above

6.

Question 6

What are the ways to create an empty object in javascript?

1 point

- `var student = new Object();`
- `var obj = {};`
- All of the above
- None of the above

7.

Question 7

Which of the following is the correct method for getting the elements using their class name?

1 point

- `document.getElementsByClassName()`
- `document.getElementByClass()`
- `document.getElementByClassName()`
- `document.getElementsByClass()`

8.

Question 8

Which of the following options is true about JavaScript?

1 point

It is an Interpreted Language

It is designed to execute Query related to DB on Server

It adds interactivity to the HTML Pages

Option 1 and 2

9.

Question 9

Functions that take other functions as arguments are known as _____.

1 point

Callback Functions

Asynchronous Functions

Anonymous functions

HigherOrder Functions

10.

Question 10

Which function of an Array object calls a function for each element in the array?

1 point

push()

forEach()

forEvery()

each()

11.

Question 11

Predict the output of the following code snippet:

```
console.log('3' + 4 + 5);
```

1 point

345

12

75

None of the above

12.

Question 12

Predict the output of the following code snippet:

```
'hi, welcome to java'.replace('java', 'javascript');
```

1 point

javascript

java

hi,welcome to java

hi,welcome to javascript

13.

Question 13

Predict the output of the following code snippet:

```
assert.lengthOf(new Map([['a',1],['b',2],['c',3]]), 3, 'map has size of 6');
```

1 point

0

1

14.

Question 14

Predict the output of the following code snippet:

```
var foo = 'hi';

assert.exists(foo, 'hi is neither `null` nor `undefined`');
```

1 point

0

1

15.

Question 15

Predict the output of the following code snippet:

```
assert.notEqual(3, 4, 'these numbers are not equal');
```

1 point

1

0

16.

Question 16

Predict the output of the following code snippet:

```
expect([10, 20, 30]).to.be.an('array').that.includes(2);
```

1 point

1

0

17.

Question 17

Predict the output of the following code snippet:

```
expect([2, 1]).to.have.ordered.members([1, 2])
```

1 point

1

0

18.

Question 18

Identify the syntax for excluding a specific testcase.

1 point

```
describe('only this test', function () {
```

```
it('only this test', function () {
```

`it.skip('only this test', function () {`

```
describe.skip('only this test', function () {
```

19.

Question 19

Identify the syntax for excluding a specified testsuite.

1 point

```
describe('only this test', function () {
```

```
it('only this test', function () {
```

```
it.skip('only this test', function () {
```

`describe.skip('only this test', function () {`

20.

Question 20

Identify the syntax for running only an individual testcase.

1 point

```
describe('only this test', function () {
```

```
it('only this test', function () {
```

`it.only('only this test', function () {`

```
describe.only('only this test', function () {
```

Server side JavaScript with Node.js

21.

Question 21

Identify the syntax for running only a specified testsuite.

1 point

```
describe('only this test', function () {
```

```
it('only this test', function () {
```

```
it.only('only this test', function () {
```

`describe.only('only this test', function () {`

22.

Question 22

Identify the Hooks provided by Mocha.

1 point

before,after, beforeclass, afterclass

before,after

before,after,beforeeach,aftereach

beforeeach,aftereach