

What is the output of following?

```
//Assume Car and Vehicle are two classes that have been already implemented.  
Car.prototype.getCar = function(){  
    console.log('Hello');  
}
```

```
Car.prototype = Vehicle.prototype;  
const car = new Car();  
car.getCar();
```

Issue arises because the Car.prototype is being reassigned (with Vehicle.prototype ) after the getCar method is added to it. Overwriting Car.prototype with Vehicle.prototype means that Car will no longer have its own prototype chain and will inherit from Vehicle instead. Since Vehicle does not have a getCar() method, then instances of Car will not have access to this method, leading to a TypeError when trying to call car.getCar(). Also here both Car and Vehicle is not defined this will also lead to an ReferenceError.

Select one:

- a. An error will be thrown since there is no method as getCar().
- b. No output.
- c. Code will execute and print 'Hello'.
- d. None of the above.



# NetExam

Sri Lanka Institute of Information Technology

In JavaScript, what does "\_\_proto\_\_" in an Object is pointing to?

Select one:

- a. To its class (Skelton) Prototype.
- b. There is no property as "\_\_proto\_\_".
- c. To the Object Itself.
- d. To its own unique Prototype object.

Q

Answer: A

A class (or skelton) prototype in JavaScript is the prototype object associated with a class. When you create an instance of a class, the instance's prototype is set to the class's prototype. This prototype object contains the properties and methods that are shared among all instances of the class. When you access a property or method on an instance, JavaScript first looks for it on the instance itself. If it's not found, JavaScript looks up the prototype chain, checking the instance's prototype, the prototype's prototype, and so on, until it either finds the property/method or reaches an object with a null prototype.



14

answered  
out of  
question

What is the output of the following code?

```
function add(a, b){  
    return a+b;  
}  
  
console.log(add(3.4,5));
```

Select one:

- a. 12
- b. 7
- c. Compilation error.
- d. undefined.

In JavaScript functions can be called with more arguments than they are defined to accept. The extra arguments are accessible within the function through the arguments object, which is an array-like object containing all the arguments passed to the function.

In a REST service developed for a library has a route for creating books. What should be the URI for adding a new book for a given author?  
(Select the best suited one)

Select one:

- a. /v1/author/{author ID}/book/
- b. /v1/books?author={author ID}
- c. /v1/authors/{author ID}/books/
- d. /v1/authors/{author ID}/create

- RESTful URIs should refer to resources, which are nouns [d]
- Use plural nouns [a, b]
- Avoiding Verbs. [d]
- RESTful APIs should use URIs to represent resources and their relationships, not query parameters to filter resources. [b]

Next page



In a REST service developed for a library has a route for creating books. What should be the URI for adding a new book for a given author? (Select the best suited one)

Select one:

- a. /v1/authors/{author ID}/books/
- b. /v1/author/{author ID}/book/
- c. /v1/books?author={author ID}
- d. /v1/authors/{author ID}/create

Next page



# NetExam

Sri Lanka Institute of Information Technology

What is the output of the following code?

```
function add(a, b){  
    return a+b;  
}  
  
console.log(add(3,4,5));
```

Select one:

- a. undefined.
- b. 7
- c. Compilation error.
- d. 12



Question 24

Not yet answered

Marked out of

1.00

Flag question

What is a code review? Select the best answer.

---

Select one:

- a. Test the code manually and using tools.
  - b. Use a tool to analyze the code.
  - c. Collaboratively check the code for potential errors and design compliance.
  - d. Build code continuously to check if there are any build failures.
-



What is the output of the following code?

```
function add(a, b){  
    return a+b;  
}  
  
console.log(add(3,4,5));
```

Select one:

- a. undefined.
- b. 12
- c. Compilation error.
- d. 7

What is the main way of creating a static variable in JavaScript?

Select one:

- a. Define a variable inside the class with the 'var' keyword.
- b. Add variable to class (Constructor function) itself.
- c. Add the variable to 'this'.
- d. Use the 'static' keyword.



Nex

What is/are the correct statement(s) about promises in JavaScript?

1. There is no advantage of using promises over callbacks.
2. Promises make the code more readable.
3. A promise can chain asynchronous tasks to work in a synchronous manner.
4. Promises can make asynchronous (ex: perform NodeJS read file synchronously) operations synchronous.

Answer: B

Select one:

- a. 1
- b. 2 and 3
- c. 2, 3 and 4
- d. 3

- **Statement 1** is incorrect because promises offer several advantages over callbacks, such as avoiding callback hell, providing a cleaner and more manageable way to handle asynchronous operations, and supporting chaining, which allows for the sequential execution of asynchronous tasks.
- **Statement 2** is correct because promises provide a more succinct and clear way of representing sequential asynchronous operations in JavaScript.
- **Statement 3** is correct because promises allow for the chaining of asynchronous operations, making it possible to execute multiple asynchronous tasks in a sequence where each subsequent operation starts when the previous operation succeeds. This makes the code more readable and easier to manage compared to using callbacks.
- **Statement 4** is incorrect because promises do not make asynchronous operations synchronous. Promises provide a way to handle the results of asynchronous operations once they complete, but the operations themselves remain asynchronous.



What is the **main** difference between a JavaScript object and a JSON object?

1. There is no difference.
2. JavaScript objects can have methods and JSON cannot.
3. In JSON attribute names (Object keys) should always be a String.

Select one:

- a. 1
- b. 1 and 3
- c.  
2 and 3
- d.  
2

**Statement 1** is incorrect because there is a significant difference between JavaScript objects and JSON objects.

**Statement 2** is correct because JavaScript objects can have methods, whereas JSON objects cannot.

**Statement 3** is correct because in JSON, all attribute names (or keys) must be strings enclosed in double quotation marks. But in JavaScript objects, keys can be strings, numbers, or identifiers without the need for quotation marks.



Question 1

Not yet answered

Marked out of  
1.00

Flag question

If you want to create a git managed repository from scratch what should be the approach?

Select one:

- a. Pull from repository.
- b. Create a remote repository and clone it or init a local repository.
- c. Push to the repository.
- d. Commit to the repository.



Question 27

Not yet answered

Marked out of  
1.00

Flag question

What is the best suited Verb and URI for creating a user in a REST API?

Select one:

- a. PUT /v1/users
- b. POST /v1/users
- c. CREATE /v1/users
- d. POST /v1/users/create



Question 4

Not yet answered

Marked out of  
1.00

Flag question

What is the the most accurate difference in arrow functions and normal functions in JavaScript?

Select one:

- a. No difference
- b. Arrow functions cannot be used with function pointers (a variable whose value is a function).
- c. Normal functions have 'this' pointing to the execution context and arrow function 'this' pointing to the surrounding context.
- d. Number of variables that can be passed to an arrow function is limited

- **Option b** is incorrect because arrow functions can indeed be used with function pointers

Next page

- **Option d** is incorrect because there is no inherent limitation on the number of variables that can be passed down to an arrow function.



## Question 25

Not yet answered

Marked out of  
1.00

Flag question

If you have a class that fetches data from the database and returns the data in HTML format. What is the principle you are violating?  
Refer to the following pseudo-code.

```
function getDate(){  
    const connection = createConnection(connectionString);  
    const data = connection.getDate();  
    return convertToHtml(data);  
}
```

Select one:

- a. Liskov substitution principle.
- b. Interface segregation principle.
- c. Class content violation principle.
- d. Single responsibility principle.

Next page



Question 20

Not yet answered

Marked out of

0

Flag question

What does it mean by static code analysis?

Select one:

- a. Testing the code manually.
- b. Checking the code for errors by another developer.
- c. Scan code using a tool for any potential erroneous scenarios and bad practices.
- d. Testing the code manually and automatically.

Question 21

Not yet answered

Marked out of  
0.00

Flag question

JavaScript (Node JS) is **best suited for?**

Select one:

- a. Computational applications
- b. Image processing applications
- c. I/O heavy applications
- d. None of the above

## Question 11

Not yet answered

Marked out of  
1.00

Flag question

What is the output of the following code?

```
function Vehicle(model){  
    this.model=model;  
}  
  
function Car(model){  
}  
  
Car.prototype = Object.create(Vehicle.prototype);  
Car.prototype.constructor = Car;  
var car = new Car('car');  
console.log(car.model);
```

Select one:

- a. Compilation error.
- b. car
- c. undefined
- d. None of the above

- Answer: A

- An error will be thrown since there is no method as getCar()

## Explanation

Issue arises because the Car.prototype is being reassigned (with Vehicle.prototype ) after the getCar method is added to it.

Overwriting Car.prototype with Vehicle.prototype means that Car will no longer have its own prototype chain and will inherit from Vehicle instead.

Since Vehicle does not have a getCar() method, then instances of Car will not have access to this method, leading to a TypeError when trying to call car.getCar().

Also here both Car and Vehicle is not defined this will also lead to an ReferenceError.



Question 30

Not yet answered

Marked out of

0

Flag question

Which conditions will return true in JavaScript?

- 1. `1 == "1"` ✓
- 2. `1 === "1"` ✗
- 3. `1 === 1` ✗
- 4. `1 == 1` ✓

Select one:

- a. 1 and 4
- b. 1, 3 and 4
- c. 3 and 4
- d. 4



Question 19

Not yet answered

Marked out of  
1.00

Flag question

What is/are the correct statement(s) about promises in JavaScript?

1. There is no advantage of using promises over callbacks. X
2. Promises make the code more readable. ✓
3. A promise can chain asynchronous tasks to work in a synchronous manner. ✓
4. Promises can make asynchronous (ex: perform NodeJS read file synchronously) operations synchronous. X

Select one:

- a. 1
- b. 2, 3 and 4
- c. 2 and 3
- d. 3



# NetExam

Sri Lanka Institute of Information Technology

on 21

et answered

ed out of

ng question

What does higher-level modules (in dependency inversion principle)?

Select one:

- a. DB level modules.
- b. UI level modules.
- c. Business logic related modules.
- d. None of the above.





Question 5

Not yet answered

Marked out of  
1.00

Flag question

You have developed a web application using Node JS. Recently a requirement raised to create a rules engine that needs heavy CPU power. What will be your optimum solution?

Select one:

- a. Drop the requirement and let them know that it is not feasible.
- b. Implement it using Node JS without any difference.
- c. Rewrite the application in Java.
- d. Try it with the latest Node JS features such as workers.

Next page



14

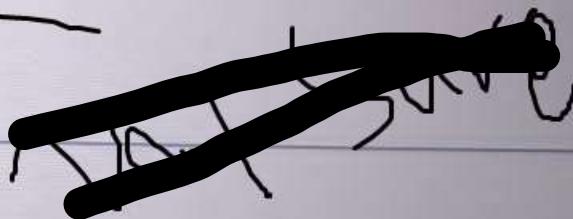
answered  
out of

question

In an application that connects to a DB cluster with replication, an issue has been detected where the data that has been written cannot be fetched immediately. But after few seconds data is available. What may be causing this (most likely)?

Select one:

- a. Error in the DB driver used in the application.
- b. DB replication lag (delay in writing the data to replicas/eventual consistency). 
- c. Error in the application logic.
- d. None of the above.



Next page



What is the reason setTimeout method always execute on or after the given millisecond value? Mark the correct statement.

Select one:

- a. If you pass 0 as the timeout value, it will execute right away.
- b. There is no function as such.
- c. Irrespective of the millisecond value setTimeout function always use event loop and main thread cannot be interrupted.
- d. All above is incorrect

Question 28

Not yet answered

Marked out of

0

Flag question

What does it mean by static code analysis?

Select one:

- a. Scan code using a tool for any potential erroneous scenarios and bad practices.
- b. Testing the code manually and automatically.
- c. Testing the code manually.
- d. Checking the code for errors by another developer.



in 22

answered

1 out of

question

What is the output of the following code?

```
window.name = 'Firefox';
function printName() {
    console.log(this.name);
}

const chrome = (
    name: 'Chrome',
    print: printName
);

printName();
chrome.print();
```

Select one:

- a. Firefox, Chrome
- b. Chrome, Chrome
- c. Firefox, Firefox
- d. The code will not run.



Question 6

Not yet answered

Marked out of

1.00

Flag question

When you are creating a class in JavaScript if you have the requirement to create multiple instances from that class what will be the best way you can do that?

Select one:

- a. Using a constructor function and adding all the variables to 'this' and behaviors to prototype.
- b. We cannot meet the above requirement in JavaScript.
- c. Using a constructor function and adding all the variables and behaviors to 'this'
- d. Using object literals ({}).



Not survl

Next page

**Answer: A**

**Option C** - Less memory-efficient than using the prototype to share methods among instances.

**Option D** - Object literals ('{}') are considered to be singleton



Question 29

Not yet answered

Marked out of  
1.00

Flag question

In an application that connects to a DB cluster with replication, an issue has been detected where the data that has been written cannot be fetched immediately. But after few seconds data is available. What may be causing this (most likely)?

Select one:

- a. Error in the DB driver used in the application.
- b. DB replication lag (delay in writing the data to replicas/eventual consistency).
- c. Error in the application logic.
- d. None of the above.

Next page



Question 27

Not yet answered

Marked out of  
0

Flag question

In JavaScript, what does "`__proto__`" in an Object is pointing to?

Select one:

- a. To the Object itself.
- b. To Its own unique Prototype object.
- c. There is no property as "`__proto__`".
- d. To its class (Skelton) Prototype. ✓

Answer: D

A class (or skelton) prototype in JavaScript is the prototype object associated with a class. When you create an instance of a class, the instance's prototype is set to the class's prototype. This prototype object contains the properties and methods that are shared among all instances of the class. When you access a property or method on an instance, JavaScript first looks for it on the instance itself. If it's not found, JavaScript looks up the prototype chain, checking the instance's prototype, the prototype's prototype, and so on, until it either finds the property/method or reaches an object with a null prototype.





### Question 6

Not yet answered

Marked out of  
1.00

Flag question

What is the output of following?

//Assume Car and Vehicle are two classes that have been already implemented.

```
Car.prototype.getCar = function(){  
    console.log('Hello');  
}
```

```
Car.prototype = Vehicle.prototype;  
const car = new Car();  
car.getCar();
```



Select one:

a.

An error will be thrown since there is no method as getCar().

b. No output.

c. Code will execute and print 'Hello'.

d. None of the above.



**Question 12**

Not yet answered

Marked out of  
1.00

Flag question

What is/are the correct statement(s) about promises in JavaScript?

1. There is no advantage of using promises over callbacks. X
2. Promises make the code more readable. ✓
3. A promise can chain asynchronous tasks to work in a synchronous manner. ✓
4. Promises can make asynchronous (ex: perform NodeJS read file synchronously) operations synchronous. X

Select one:

- a. 2, 3 and 4
- b. 2 and 3
- c. 3
- d. 1



## Question 30

Not yet answered

Marked out of  
1.00

Flag question

What is the usage of the ".gitignore" file?



Select one:

- a. A list of node\_modules that should be ignored for tracking.
- b. A list of all the files and directories that should be ignored for tracking.
- c. A list of all the branches that should be ignored for tracking.
- d. To keep some metadata about the repository.



Finish attempt...

**Question 17**

Not yet answered

Marked out of  
1.00

Flag question

If you want to create a git managed repository from scratch what should be the approach?

Select one:

- a. Create a remote repository and clone it or init a local repository.
- b. Pull from repository.
- c. Commit to the repository.
- d. Push to the repository.



## Question 30

Not yet answered

Marked out of  
1.00

Flag question

If you have a class that fetches data from the database and returns the data in HTML format. What is the principle you are violating? Refer to the following pseudo-code.

```
function getDate(){  
    const connection = createConnection(connectionString);  
    const data = connection.getDate();  
    return convertToHtml(data);  
}
```

Select one:

- a. Class content violation principle.
- b. Single responsibility principle.
- c. Interface segregation principle.
- d. Liskov substitution principle.



Finish attempt...



What is the best HTTP word for updating (selected set of fields not replacing the entire resource) a resource?

Select one:

- a. OPTIONS
- b. PATCH
- c. PUT
- d. POST

## Question 23

Not yet answered

Marked out of  
1.00

Flag question

What is not an asynchronous programming pattern?

Select one:

- a. Callbacks
- b. push/pop
- c. Async/await
- d. Promises



Question 8

Not yet answered

Marked out of  
1.00

Flag question

What is the reason setTimeout method always execute on or after the given millisecond value? Mark the correct statement.

Select one:

- a. If you pass 0 as the timeout value, it will execute right away.
- b. There is no function as such.
- c. Irrespective of the millisecond value setTimeout function always use event loop and main thread cannot be interrupted.
- d. All above is incorrect

Next page



Question 7

Not yet answered

Marked out of

0.00

Flag question

What is not an asynchronous programming pattern?

Select one:

- a. push/pop
- b. Async/await
- c. Promises
- d. Callbacks



**Question 20**

Not yet answered

Marked out of  
1.00

Flag question

Consider the following statements.

```
let a =  
if(a) {  
}
```

To evaluate the above condition to false, what is the possible value for variable 'a'; Answer: D

Select one:

- a. var a = 1
- b. var a = 'false'
- c. var a=""
- d. None of the above

None of the above ✓



Question 21

Not yet answered

Marked out of  
1.00

Flag question

What is a code review? Select the best answer.

Select one:

- a. Collaboratively check the code for potential errors and design compliance.
- b. Build code continuously to check if there are any build failures.
- c. Test the code manually and using tools.
- d. Use a tool to analyze the code.



Question 13

Not yet answered

Marked out of  
1.00

Flag question

What does "Closure" means in JavaScript?

Select one:

- a. Define a function in one place and pass it to anywhere and use it
- b. Completed function without any flaws
- c. The nested function ability to access its parent function context anywhere it's being used.
- d. A nested function

**Option C** is correct because a closure in JavaScript is a function that has access to its own scope, the outer function's scope, and the global scope.

Next page



**Question 3**

Not yet answered

Marked out of  
1.00

Flag question

JavaScript (Node JS) is best suited for?

Select one:

- a. Computational applications
- b. Image processing applications
- c. I/O heavy applications
- d. None of the above



You have developed a web application using Node JS. Recently a requirement raised to create a rules engine that needs heavy CPU power. What will be your optimum solution?

Select one:

- a. Try it with the latest Node JS features such as workers.
- b. Drop the requirement and let them know that it is not feasible.
- c. Rewrite the application in Java.
- d. Implement it using Node JS without any difference.

Next page