



Basics of OOPS - Quiz

Below is a scored review of your assessment. All questions are shown.



Correct Answer



Partially Correct



Incorrect Answer

1 When a class B can extend another class A, we say that?

Your Answer A is the superclass and B is the subclass

Correct Answer A is the superclass and B is the subclass

Justification

When a class B can extend another class A, we say that A is the superclass and B is the subclass.

2 How does Javascript achieve Inheritance?

Your Answer Inheritance through Prototype chaining

Correct Answer Inheritance through Prototype chaining

Justification

JavaScript achieves inheritance through prototype chaining, where a sub-class prototype object references its parent class prototype object.

3 "this" can point to global object or the object that contains the function.

Your Answer False

Correct Answer True

Justification

True, "this" can point to the global object or the object that contains the function.

- 4 When a function is used as a constructor, the newly created object inherits properties from the _____ object.

Your Answer prototype

Correct Answer prototype

Justification

When a function is used as a constructor, the newly created object inherits properties from the prototype object.

5

What will be the output of the following JavaScript code?

```
const obj1 = {  
  property1: 2  
};  
Object.seal(obj1);  
obj1.property1 = 4;  
console.log(obj1.property1);  
delete obj1.property1;
```

Your Answer

Error

Correct Answer

4

Justification

The code creates an object `obj1` with a property `property1` set to `2`. The `Object.seal(obj1)` method is used to seal the object, allowing value modifications but not addition or deletion of properties. The code then changes `obj1.property1` to `4`, which is permitted. However, the attempt to delete `property1` is not allowed due to the object being sealed. Therefore, `console.log(obj1.property1)` prints `4`, the updated value of `property1`.

6 What will be the output of the given code?

```
class Rect {  
  constructor(h, w) {  
    this.height = h;  
    this.width = w;  
  }  
  foo() {  
    return this.height * this.width;  
  }  
}  
const sq = new Rect(5, 20);  
console.log(sq.foo());
```

Your Answer 100

Correct Answer 100

Justification

The output of the code will be 100, as the foo() method returns the product of height and width of the object.

7

What will be the output of the following JavaScript code?

```
const prototype1 = {};  
const object1 = Object.create(prototype1);  
console.log(Object.getPrototypeOf(object1) === prototype1);
```

Your Answer true

Correct Answer true

Justification

The output of the JavaScript code will be "true", as the prototype of "object1" is set to "prototype1" using the `Object.create()` method, and `Object.getPrototypeOf()` method is used to check if the prototype of "object1" is equal to "prototype1".

8 The _____ method is like the call() method, except that the arguments to be passed to the function are specified as an array.

Your Answer apply()

Correct Answer apply()

Justification

The apply() method is like the call() method, except that the arguments to be passed to the function are specified as an array.

9

What will be the output of following code snippet?

```
function F() {  
    // Constructor function definition  
}  
var o = new F();  
console.log(o.constructor === F);
```

Your Answer true

Correct Answer true

Justification

The constructor property of an object created with a constructor function refers to the constructor function itself.

10 Both call() and apply() methods allow you to explicitly specify the _____ value for the invocation.

Your Answer this

Correct Answer this

Justification

The call() and apply() methods allow you to explicitly specify the "this" value for the invocation.

Page 1 of 1

Summary

I'm done.

Software by

Version 11.2

Privacy Policy. Assessment content is copyright 2024, AlmaBetter.