

JavaScript Submission 3

1. What is “closure” in javascript? Can you provide an example?
2. What are promises and how are they useful?
3. How to check whether a key exists in a JavaScript object or not.
4. What is the output of this code? **Please explain**

```
var employeeId = 'abc123';

function foo() {
    employeeId();
    return;

    function employeeId() {
        console.log(typeof employeeId);
    }
}

foo();
```

5. What is the output of the following? **Please explain**

```
(function() {  
    'use strict';  
  
    var person = {  
        name: 'John'  
    };  
    person.salary = '10000$';  
    person['country'] = 'USA';  
  
    Object.defineProperty(person, 'phoneNo', {  
        value: '8888888888',  
        enumerable: true  
    })  
  
    console.log(Object.keys(person));  
})();
```

6. What is the output of the code? Explain

```
(function() {  
    var objA = {  
        foo: 'foo',  
        bar: 'bar'  
    };  
    var objB = {  
        foo: 'foo',  
        bar: 'bar'  
    };  
    console.log(objA == objB);  
    console.log(objA === objB);  
})();
```

7. What is the output of the following code:

```
function Person(name, age){  
    this.name = name || "John";  
    this.age = age || 24;  
    this.displayName = function(){  
        console.log(this.name);  
    }  
}  
  
Person.name = "John";  
Person.displayName = function(){  
    console.log(this.name);  
}  
  
var person1 = new Person('John');  
    person1.displayName();  
    Person.displayName();
```

8. In-Class Exercise: Designing a School Management System

Scenario:

You are tasked with designing a School Management System for a school. The system should manage students, teachers, courses, and their interactions.

Exercise Instructions:

1. Identify Classes:

- List down the main entities (classes) that you think are necessary for the School Management System. Consider entities like **Student**, **Teacher**, **Course**, etc.

2. Define Class Properties:

- For each identified class, define the properties (attributes) that would be essential to store information. For example, **Student** class might have properties like **id**, **name**, **email**, etc.

3. **Define Class Methods:**

- Specify the methods (functions) that each class should have. Think about what actions each class needs to perform. For instance, **Student** might need methods like `enroll(course)`, `getGrades()`, etc.

4. **Class Relationships:**

- Determine how classes will interact with each other. For example, how will a **Teacher** assign a **Course** to a **Student**? How will a **Course** keep track of enrolled **Students**?

5. **Write Sample Code:**

- Write a basic implementation in JavaScript using classes and methods you've defined. This step can help reinforce understanding through practical application.