|  |
| --- |
| Part 1: Multiple Choices Questions |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1. **[1 point]** The main purpose of a Lexical enviroment is to contain:  |  | | --- | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | A. the code inside that environment and reference to surrounding environment |  | | B. variables inside that environment and reference to surrounding environment |  | | C. a dictionary of keywords and their meaning for a program |  | | |
|  | 1. **[1 point]** Lexical environment is another term for execution context  |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | | True | | False | | |

Part 2: Short Answer Questions

**NOTE: In the real exam, I may ask you function with parmaters/arguments and closure too.**

 3. **[8 points]** Read the code below:

1) What's the global Lexical Environment in creation phase?

2) What's the global Lexical Environment in execution phase?

3)  What's the lexical environment for function b in creation phase?

4) What's the lexical environment for function b in execution phase?

 function a(x) {

  let y = 10;

  function f() {

    y = 20;

  }

  f();

  console.log(x);

}

function b() {

  const x = 10;

  a(30);

}

const x = 20;

b();

4. **[2 points]** Read code below and answer questions below:

1. List \_\_proto\_\_ and all properties with values and methods for class User only, don't include parents' ones.

2. List \_\_proto\_\_ and all properties with values and methods for john object only, don't include parents' ones.

**NOTE: In the real exam, I may ask you to List all properties with values and methods for class User.prototype. multi-level Inheritance, static properties, static methods. Also may include questions based on code using Object literal or constructor functions**

class User {

    static name = 'Default';

    constructor(name) {

        this.name = name;

    }

    get name() {

        return this.\_name;

    }

    set name(value) {

        if (value.length < 4) {

            alert("Name is too short.");

        } else {

            this.\_name = value;

        }

    }

}

let john = new User("John");

5. **[4 points]** If you execute the code below, you'll see undefined for the name in the console shown:

function askPassword(ok, fail) {

let password = prompt("Password?", '');

if (password == "rockstar") ok();

else fail();

}

let user = {

name: 'John',

loginOk() {

console.log(`${this.name} logged in`);

},

loginFail() {

console.log(`${this.name} failed to log in`);

},

};

askPassword(user.loginOk, user.loginFail);

Fix the problem using 4 ways: wrapper function, call, apply and bind.

**NOTE: In the real exam, you need to understand self pattern like homework. This question is not a good example to demo self pattern.**

Part 3: Programming Questions

6. **[5 points]** Add a new method named mysort() without parameters in built-in constructor function Array. It’ll sort all elements in the array in descending order.

After completion of the method, we can call as below. mysort() doesn't return a new array, it modifies on original array

let arr = [7, 5, 2, 4, 3, 9];

arr.mysort();

console.log(arr); //Expected result: [ 9, 7, 5, 4, 3, 2 ]

**NOTE: In the exam, I'll ask you similar questions related to native prototypes. It's not limited in Array, may ask you create a new method in String, Object, etc. Only 1 method though.**

7. **[10 points]** Use constructor function or class syntax to create a new class named LinkedList. Find out all usage of constructor in code below:

The methods of LinkedList are below:

1) add(item): add a item to the linked list.

2) remove(value): remove a item to the linked list.

You’re allowed to add other methods if it’s necessary.

After completion, we can call methods as below and see the results in console if call print(). The print() method is already given, no need to implement.

let linkedlist = new LinkedList();

linkedlist.add(1);

linkedlist.add(2);

linkedlist.add(3);

linkedlist.print(); //in the console, you should see: LinkedList{1,2,3}

linkedlist.remove(3);

linkedlist.print(); //in the console, you should see: LinkedList{1,3}

let linkedlist2 = new LinkedList([1,2,3]);

linkedlist2.print(); //in the console, you should see: LinkedList{1,2,3}

 You're allowed to use Array APIs to do this question.

**NOTE: In the real exam, I may ask you to use class syntax or constructor function to implement data structures we have learnt from this course, it won't be Array, but may be Set, LinkedList, Stack and Queue. And you have to read the given code to find out constructor, methods, properties, etc by yourself. I'll listed all methods needed to be implemented, but constructor and properties, you have to find out by yourself.**

8. **[15 points]** This is a quiz system which allows students to take quiz, get each student's quiz score and compute average score of students.

You need to use constructor functions or class syntax to implement Student, Question and Quiz

constructor function/class Student:

properties:

studentId: a unique student id

answers: holds an array that records the student's answers for the questions.

method: addAnswer(question) - add student's question(id, answer) to answers array.

constructor function/class Question:

properties:

qid: unique quesiton id

answer: hold quiz correct answer or student's answer

method: checkAnswer(answer) - used to check if student's answer is correct

constructor function/class Quiz:

properties:

questions: It's a Map which holds question id and correct answers. The key is question id, the value is the correct answer for this question

students: an array holds all students

methods:

* scoreStudent(sid), computes the quiz score for this student
* getAverageScore(), computes the average score over all students

After you complete the Question, Student and Quiz constructor functions, we may use the system as below:

Your system should return the correct result for getAverageScore() and scoreStudent(sid) as the expected result.

const student1 = new Student(10);

student1.addAnswer(new Question(2, 'a'));

student1.addAnswer(new Question(3, 'b'));

student1.addAnswer(new Question(1, 'b'));

const student2 = new Student(11);

student2.addAnswer(new Question(3, 'b'));

student2.addAnswer(new Question(2, 'a'));

student2.addAnswer(new Question(1, 'd'));

const students = [student1, student2];

const questions =[new Question(1, 'b'), new Question(2, 'a'), new Question(3, 'b')];

const quiz = new Quiz(questions, students);

let scoreforStudent10 = quiz.scoreStudentBySid(10);

console.log(scoreforStudent10); //Expected Result: 3

let scoreforStudent11 = quiz.scoreStudentBySid(11);

console.log(scoreforStudent11); //Expected Result: 2

let average = quiz.getAverageScore();

console.log(average); //Expected Reuslt: 2.5