Answersheet

Name: Arashad Ahamad **No. of questions attempted:** 7 **Submitted at:** 19 May, 2025 7:01 PM

Assignment: Execution Context in JavaScript **Total no. of questions:** 7 **Total marks:** 15

Result: Not reviewed yet

Sl. No.	Question	Actions
1	**What is an execution context in JavaScript, and what are the different types of execution contexts?** 5 marks What is an Execution Context in JavaScript? An Execution Context is the environment in which JavaScript code is evaluated and executed. It contains everything the JavaScript engine needs to run code properly, like: Variable environment (variables, functions) Scope chain this keyword 🧠 Types of Execution Context in JavaScript 1. Global Execution Context (GEC) Created when the JavaScript code starts running. Only one global context exists. Code not inside any function runs here. In the browser, window is the global object. var a = 10; // runs in Global Execution Context 2. Function Execution Context (FEC) Created whenever a function is called. Each function has its own execution context. A new context is pushed to the call stack when the function is invoked. function greet() { console.log("Hello"); } greet(); // creates Function Execution Context	Marks 0
2	**How is the execution context created and managed during the execution of JavaScript code?** 5 marks	Marks

	1. Creation Phase The JavaScript engine creates the Execution Context. In this phase: Memory is allocated for variables and functions. Variables are initialized with undefined. Functions are stored as a whole (their code). The value of this is determined. The Lexical Environment is set up. 2. Execution Phase JavaScript code is executed line by line. Variables get their actual values. Functions are invoked, and for each function call, a new Execution Context is created. 3. Call Stack Management All Execution Contexts are managed using the Call Stack. The Global Execution Context (GEC) is pushed first. When a function is called, a Function Execution Context (FEC) is pushed onto the stack. When the function finishes, its context is popped off the stack.	0
3	**What is the global execution context in JavaScript?** 1 mark 1. The execution context within a function 2. The execution context of a block [✓ 3. The outermost execution context, created when a script is run] 4. The execution context of a nested function	**Correct answer** 1 mark
4	**What happens when a function is used and the debugger is enabled?** 1 mark 1. The function is skipped during execution [✓ 2. The function is executed, and the debugger pauses execution at the function's entry point] 3. The function is executed, and the debugger logs messages to the console 4. The function execution speed is increased	**Correct answer** 1 mark
5	**What happens when a variable declaration inside a block using const is changed to var?** 1 mark	**Correct answer**

	1. The variable becomes globally scoped 2. The variable becomes functionally scoped [✓ 3. The variable's scope changes from block to function or global depending upon the location of that block] 4. The variable becomes immutable	1 mark
6	**What is the local execution context in JavaScript?** 1 mark [✓ 1. The execution context within a function] 2. The execution context of the main JavaScript file 3. The outermost execution context 4. The execution context of a nested function	Correct answer 1 mark
7	**What is the scope chain in JavaScript and its role within the execution context?** 1 mark 1. It refers to the order in which functions are executed within the code, ensuring synchronous behavior. [✓ 2. It determines the visibility and accessibility of variables within a particular context, facilitating variable resolution.] 3. It is a data structure that stores variables in memory, allowing for efficient retrieval during execution. 4. It represents the hierarchy of HTML elements in the DOM, enabling manipulation of webpage content dynamically.	Correct answer 1 mark