ANSWER 1-

Hoisting in JavaScript is a behavior that affects how variable and function declarations are processed during the compilation phase of the code execution. It allows you to use variables and functions before they are declared in your code, as if they were "hoisted" or lifted to the top of their respective scopes

In JavaScript, when the code is executed, it goes through two phases: the compilation phase and the execution phase. During the compilation phase, the JavaScript engine scans the code and recognizes variable and function declarations and assigns memory space for them. This is where hoisting comes into play. During hoisting, variable declarations (but not their assignments) and function declarations are moved to the top of their respective scopes. This means that you can access and use them before they are explicitly declared in the code.

It's important to note that hoisting only moves the declarations, not the initializations or assignments. In the case of variables, if you don't explicitly initialize them before the hoisted declaration, they will have the value undefined. Similarly, function expressions (assigned to variables) are not hoisted, only function declarations. To avoid confusion and potential bugs, it's generally recommended to declare variables at the beginning of their respective scopes and to define functions before using them, even though hoisting allows for flexibility in the ordering of code.