**1.) Introduction to programming**

**2.) ECMAScript**

**3.) How to execute JavaScript?**

**4.) Variables**: In JavaScript, variables are the container for storing data (storing data values). eg: x,y,z are variables declared with the var keyword.

**5.) Identifier**: It is a sequence of characters in the code that identifies a variable function or property. In, JavaScript identifiers are case-sensitive and can contain uni-code letters, $, \_, and digit (0-9), but may not start with a digit.

**6.) Rules for choosing JavaScript**

**7.)Var, Const, and Let in js:**

In JavaScript, var, const, and let are all used to declare variables. They each have their own specific use cases, which you can use to choose the right keyword for your variable declaration.

var is the oldest and most widely-used keyword for declaring variables in JavaScript. It is used to declare variables that are either global or local to the function in which they are declared. Var is globally scooped. Variables declared with var can be reassigned and can be redeclared. Thus, their value can be changed throughout the lifetime of your program.

const is used to declare variables that are constant, meaning that their value cannot be changed once they are assigned. If you try to reassign a value to a variable declared with const, you will get an error. const declarations are also block-scoped, meaning that they are only visible within the block of code in which they are declared. Also, the const can neither be updated nor re-declared.

let is similar to var, but it is block-scoped rather than function-scoped. This means that variables declared with let are only visible within the block of code in which they are declared, and cannot be accessed outside of that block. Let declarations can also be reassigned but not re-declared.

In general, it is considered best practice to use const whenever possible, and only use let or var if you need to reassign the value of the variable. This helps to prevent accidental reassignment and makes your code more readable and maintainable.