1. **Data Types and Variables**:

1.1 What are the different data types used in JavaScript variables in the provided code

• Number: Represents both integers and floating-point numbers. • String: A sequence of characters, e.g., "hello".

• Boolean: Logical values true or false.

• Undefined: A variable that has been declared but not assigned a value.

• Null: Represents the intentional absence of any object value.

• Symbol: A unique and immutable data type often used as object keys.

• Object: Used to store collections of data and more complex entities.

1.3 Explain the difference between var, let, and const in JavaScript.

• var: Declares a variable globally or function-scoped. Variables can be re-declared or updated

• let: Declares a block-scoped variable. Variables can be updated but not re-declared within the same block

• const: Declares a block-scoped constant. It cannot be re-assigned or re-declared.

1.3 Why JavaScript Allows Assigning Different Data Types to the Same Variable

JavaScript is a dynamically typed language, meaning variables are not bound to a specific type. You can change a variable’s type by simply assigning a new value of a different type.

1 JavaScript Overview

2 1.4 Handling Variables Declared but Not Initialized

A variable that is declared but not initialized has the value undefined.

Example: let x; console. log (x); // undefined

1.5 Significance of Variable Names

Variable names in programming are important for clarity, readability. In JavaScript, meaningful variable names help in understanding the purpose of the variable and its role in the program.

2 Numeric Data Types

2.1 Various Numeric Data Types

• Integer: Whole numbers like 10, -5.

• Floating-point (double): Numbers with decimal points like 3.14, -2.718.

• Infinity: A special value that results from dividing a non-zero number by 0, e.g., 10 / 0. 2.2 Difference Between Integers, Doubles, and Infinity

• Integers: Whole numbers without a fractional part.

• Doubles (floating-point): Numbers with a fractional part.

• Infinity: Represents a value larger than any number or negative infinity for the smallest value.

2.3 Handling Arithmetic Operations JavaScript handles arithmetic operations by automatically converting numeric types when necessary (e.g., adding an integer to a floating-point number).

3 String Data Type

3.1 String Representation in JavaScript Strings are sequences of characters enclosed in either single

(’. . .’) or double (". . . ") quotes.

3.2 Difference Between Single and Double Quotes

There is no functional difference between single and double quotes. They can be used interchangeably, but consistency is key.

JavaScript Overview

3 3.3 Automatic Conversion of Characters to Strings in JavaScript, any character or group of characters enclosed in quotes is treated as a string, regardless of the number of characters.

4.0 Boolean and Undefined Data Types

4.1 Purpose of Boolean Variables Boolean variables represent logical entities and can have two values: true or false. They are useful for controlling flow in conditions.

4.2 Concept of Undefined A variable that is declared but not assigned a value will have the value undefined.

Example: let y; console. log (y); // undefined

4.3 Use of Booleans in Conditional Statements Boolean variables are critical in conditional logic, enabling the execution of code blocks based on true or false values.

Example: let is Valid = true; if (is Valid) {console. log (’ Valid!’); } 5 Null Data Type

5.1 Significance of null null is used to represent an intentional absence of any object value. It’s a placeholder for an object that doesn’t yet exist.

5.2 Difference Between null and undefined

• undefined: A variable has been declared but has no value.

• null: An assignment value that represents” no value” intentionally.

Example: let x = null; console. log (x); // null JavaScript Overview

6.0 Object Data Type

6.1 How Objects are Represented in JavaScript

Objects are collections of key-value pairs, where the keys are strings (or Symbols), and the values can be any data type.

Example: let person = {name: " John “, age: 30};

6.2 Structure of the country Info Object

Example: let country Info = {name: " Kenya ", capital: " Nairobi “};

6.3 Nesting Objects Objects can contain other objects as values.

/Example: let country Info = {name: " Kenya ", capital: " Nairobi “, population: {males: 1000000, females: 1100000};

7.0 Array Data Type

7.1 Purpose and Structure of Arrays Arrays store ordered lists of values, which can be any data type. Example: let fruits = [" apple “, " banana “, " orange "];

7.2 Arrays with Different Data Types

Arrays can hold multiple data types.

JavaScript Overview 5

Example: let mixed Array = [42, " hello “, true, null];

7.3 Array of Arrays (2D Arrays) Arrays can contain other arrays, allowing for multi-dimensional data representation.

Example: let matrix = [ [1, 2, 3], [4, 5, 6], [7, 8, 9];

8.0 Variable Naming Conventions

8.1 Naming Conventions

• Variable names can contain letters, digits, underscores, and dollar signs.

• They must begin with a letter, underscore, or dollar sign.

• Camel case is commonly used in JavaScript (myVariableName).

8.2 Importance of Descriptive Names Descriptive names make code more readable and maintainable. It’s good practice to choose names that convey the purpose of the variable.

8.3 Naming Conventions Followed or Violated Ensure that variables are properly named following camel case and not starting with digits.

9.0 Constants in JavaScript

9.1 Use of const Keyword

Constants (const) are variables that cannot be reassigned. They are useful when the value should remain constant throughout the program.