### **Variables and Data Types Questions**

1. **Declaring and Initializing Variables**:  
   Declare variables for a person's first name, last name, age, and city. Assign values and print them in separate lines.
2. **Using Constants**:  
   Define a constant to store the value of pi (3.14). Try to reassign a new value to it and explain the error.
3. **Type Conversion**:  
   Create a variable with a string value of "123". Convert it to a number and demonstrate the type conversion using typeof.
4. **String Length**:  
   Create a variable with a string value. Write a program to find and print the length of the string.
5. **Character Access**:  
   Declare a string variable containing your favorite word. Write a program to print the first and last characters of the word.
6. **Arithmetic with Variables**:  
   Declare two variables and assign them numeric values. Perform operations to find their sum, difference, product, and quotient. Print each result.
7. **Boolean Evaluation**:  
   Create two boolean variables, one with true and one with false. Write a program to print the result of AND, OR, and NOT operations on them.
8. **String Methods**:  
   Declare a variable with a sentence like, "JavaScript is fun!". Use string methods to:
   * Convert it to uppercase.
   * Replace "fun" with "awesome".
   * Check if the string includes the word "JavaScript".
9. **Undefined Variables**:  
   Declare a variable without assigning a value. Print its value and type. Then assign a number to it and print the updated value and type.
10. **Swapping Variables**:  
    Declare two variables with numeric values. Swap their values without using a temporary variable and print the final result.

### **Conditional Logic Questions**

1. **Basic If-Else**:  
   Write a program that checks if a number stored in a variable is positive. If it is positive, print "Positive", otherwise print "Not Positive".
2. **Odd or Even**:  
   Create a program that checks if a number is odd or even using an if-else statement. Print the result.
3. **Grade Categorization**:  
   Write a program that takes a number (0-100) and prints the grade category:
   * "A" for 90-100
   * "B" for 80-89
   * "C" for 70-79
   * "D" for 60-69
   * "F" for below 60
4. **Multiple Conditions**:  
   Check if a person is eligible to vote. The person must be at least 18 years old and have valid citizenship. Print "Eligible" or "Not Eligible".
5. **Using Ternary Operators**:  
   Write a program that checks if a number is divisible by 5. Use a ternary operator to print "Divisible by 5" or "Not Divisible by 5".
6. **Nested If-Else**:  
   A student needs at least 50% in each of three subjects to pass an exam. Write a program that takes marks for the three subjects and prints "Pass" if all conditions are met, otherwise "Fail".
7. **Switch Statement**:  
   Create a program that takes the day of the week (as a number 1-7) and prints the name of the day (e.g., 1 = "Monday", 2 = "Tuesday"). Use a switch statement.
8. **Complex Conditions**:  
   Write a program that checks if a year is a leap year. A leap year is divisible by 4 but not by 100, unless it is also divisible by 400.
9. **If-Else with Range Checking**:  
   A ticketing system charges based on age:
   * Free for ages 0-5
   * $10 for ages 6-17
   * $20 for ages 18-59
   * $15 for ages 60 and above  
     Write a program that takes age as input and prints the corresponding ticket price.
10. **Nested Switch Cases**:  
    Write a program that takes a vehicle type (car, motorcycle, or truck) and fuel type (petrol, diesel) as input. Use nested switch cases to print "Allowed" or "Not Allowed" based on these rules:
    * Cars and motorcycles using petrol are allowed.
    * Trucks using diesel are allowed.
    * Any other combination is not allowed.

### **Loops Questions**

1. **Print Numbers with a For Loop**:  
   Write a program that uses a for loop to print numbers from 1 to 10.
2. **Sum of Numbers**:  
   Write a program that uses a for loop to calculate and print the sum of numbers from 1 to 100.
3. **Print Even Numbers**:  
   Use a while loop to print all even numbers between 1 and 20.
4. **Factorial Calculation**:  
   Write a program to calculate the factorial of a number using a for loop. For example, if the input is 5, the output should be 120 (5 x 4 x 3 x 2 x 1).
5. **Reverse Counting**:  
   Use a do-while loop to print numbers from 10 down to 1.
6. **Multiplication Table**:  
   Write a program that prints the multiplication table of a given number (e.g., for 5, it should print 5 x 1 = 5, 5 x 2 = 10, etc., up to 5 x 10).
7. **Sum of Odd Numbers**:  
   Use a for loop to calculate the sum of all odd numbers between 1 and 50. Print the result.
8. **Nested Loops for Patterns**:  
   Write a program to print the following pattern using nested for loops:

\*

\*\*

\*\*\*

\*\*\*\*\*

1. **Finding the Largest Number in an Array**:  
   Given an array of numbers, use a loop to find and print the largest number in the array.
2. **Prime Numbers**:  
   Write a program that uses a loop to find and print all prime numbers between 1 and 50.

### **Objects Questions**

1. **Basic Object Creation**:  
   Create an object to store information about a book, including properties like title, author, and yearPublished. Print the object.
2. **Accessing Properties**:  
   Write a program that accesses and prints the title and author properties from a given book object.
3. **Updating Properties**:  
   Given an object representing a car with properties like make, model, and year, write a program to update the year to the current year and print the updated object.
4. **Adding Properties**:  
   Create an object to store a student's information (e.g., name, age). Add a new property grade to the object and assign it a value. Print the updated object.
5. **Deleting Properties**:  
   Create an object with properties name, age, and city. Delete the city property and print the updated object.
6. **Iterating Over an Object**:  
   Write a program to iterate over the properties of an object using a for...in loop and print each key and its value.
7. **Object with Methods**:  
   Create an object representing a rectangle with properties width and height. Add a method calculateArea that returns the area of the rectangle. Call the method and print the result.
8. **Array of Objects**:  
   Create an array of objects, where each object represents a student with properties name and grade. Use a loop to print the name of each student who scored above 80.
9. **Dynamic Property Access**:  
   Write a program that takes a property name as input and prints the corresponding value from a given object. If the property doesn’t exist, print a default message like "Property not found."
10. **Merging Objects**:  
    Write a program to merge two objects into a single object. For example:  
    const obj1 = { a: 1, b: 2 }  
    const obj2 = { c: 3, d: 4 };  
    const obj3 (merged object) = { a: 1, b: 2, c: 3, d: 4 }

### **Functions Questions**

1. **Basic Function Declaration**:  
   Write a function named greet that takes a name as a parameter and prints "Hello, [Name]!".
2. **Function with Return Value**:  
   Create a function named square that takes a number as input and returns its square.
3. **Default Parameters**:  
   Write a function named calculateArea that calculates the area of a rectangle. Use default values for width and height if no arguments are provided.
4. **Iterative Function**:  
   Write a function that takes an array of numbers and returns the sum of all numbers in the array using an iterative approach.
5. **Recursive Function**:  
   Create a recursive function named factorial that calculates the factorial of a given number.
6. **Function as a Callback**:  
   Write a function named processArray that takes an array and a callback function. The callback function should be applied to each element in the array and double the elements. For example, processArray([1, 2, 3], callback) should return [2, 4, 6].
7. **Higher-Order Function I**:  
   Create a function named filterNumbers that takes an array of numbers and another function as a filter condition; in this case, only retain numbers that are prime. The filterNumbers function should return an array of numbers that satisfy the filter condition.
8. **Higher-Order Function II**:  
   Write a function named sumEven that takes an array of numbers, a boolean, and another function as a filter condition. The sumEven function should return the sum of all exclusively even numbers in the array if the boolean is true; if the boolean passed in is false, it should return the sum of exclusively all odd numbers in the array.