Operators in JavaScript(Part-2)

Assignment Questions.

Problem Statement 1:

Explain the usage of ternary operator with syntax and write a program to check whether the number is even or odd using the ternary operator.

Ans1:

In JavaScript, the ternary operator is used to write concise conditional expressions. Its syntax is:

```
condition ? expression1 : expression2
```

If the condition evaluates to true, `expression1` is executed; otherwise, `expression2` is executed.

Here's how you can use the ternary operator to check whether a number is even or odd in JavaScript:

```
// Program to check whether a number is even or odd using the ternary operator

const number = parseInt(prompt("Enter a number: "));

const result = number % 2 === 0 ? "Even" : "Odd";

console.log(`The number ${number} is ${result}.`);
```

In this program:
- We use `prompt()` to get input from the user, then parse it into an integer using `parseInt()`.
- We use the ternary operator to check if the number modulo 2 is equal to 0. If it is, the number is even; otherwise, it's odd.
- The result is stored in the `result` variable.
- We then log the result to the console.

Problem Statement 2:

Describe the usage of the comma operator in JavaScript and provide an example.

Ans2:

In JavaScript, the comma operator allows you to evaluate multiple expressions, separated by commas, and returns the value of the last expression. It's typically used in situations where multiple expressions need to be evaluated, but only one value is needed. The expressions are evaluated from left to right, and the value of the last expression becomes the value of the entire comma-separated expression.

The syntax for the comma operator is as follows:

```
expression1, expression2, ..., expressionN
```

Here, `expression1` through `expressionN` are any valid JavaScript expressions.

The comma operator can be used in a variety of situations, such as in variable declarations, loop headers, or function arguments, where multiple expressions need to be evaluated.

Here's an example of how the comma operator can be used in JavaScript:

```
// Using comma operator in a for loop
for (var i = 0, j = 10; i < j; i++, j--) {
   console.log("i:", i, "j:", j);
}

// Output:
// i: 0 j: 10
// i: 1 j: 9
// i: 2 j: 8
// i: 3 j: 7
// i: 4 j: 6
```

In this example:
- We're using the comma operator in the initialization and increment parts of the `for` loop.
- The loop initializes two variables `i` and `j` to 0 and 10, respectively.
- The loop continues as long as `i` is less than `j`.
- In each iteration of the loop, both `i` is incremented by 1 and `j` is decremented by 1.
- The comma operator allows us to perform multiple operations within each part of the `for` loop header.

Problem Statement 3:

Use a nested ternary operator to check that a number is positive, negative or zero. You have to print “positive” if the number is positive and similarly for negative and zero also.
Ans3:
You can use a nested ternary operator to check whether a number is positive, negative, or zero, and then print the respective message. Here's how you can do it in JavaScript:

```
const number = parseInt(prompt("Enter a number: "));

const result = number > 0 ? "positive" : number < 0 ? "negative" : "zero";

console.log(`The number is ${result}.`);
```

In this code:
- We use `prompt()` to get input from the user, then parse it into an integer using `parseInt()`.
- We use a nested ternary operator to check the value of the number:
  - If the number is greater than 0, it's positive.
  - If the number is less than 0, it's negative.
  - If neither condition is true (i.e., the number is 0), it's zero.
- Depending on the result of the nested ternary operator, the appropriate message is assigned to the `result` variable.
- Finally, we log the result to the console.

Problem Statement 4:

Use a ternary operator to check that a person is eligible to vote or not by checking his age. If the age of the person is less than 18 then “You cannot vote” should be logged else “You can vote” should be logged.

Ans4:

You can use a nested ternary operator to check whether a number is positive, negative, or zero, and then print the respective message.
Here's how you can do it in JavaScript:

```
const number = parseInt(prompt("Enter a number: "));

const result = number > 0 ? "positive" : number < 0 ? "negative" : "zero";

console.log(`The number is ${result}.`);
```

In this code:
- We use `prompt()` to get input from the user, then parse it into an integer using `parseInt()`.
- We use a nested ternary operator to check the value of the number:
  - If the number is greater than 0, it's positive.
  - If the number is less than 0, it's negative.
  - If neither condition is true (i.e., the number is 0), it's zero.
- Depending on the result of the nested ternary operator, the appropriate message is assigned to the `result` variable.
- Finally, we log the result to the console.