Module 28 Assignment

Q.1- Explain the role of operators in javascript. Why are they essential in programming?
Ans- Operators in JavaScript are symbols or keywords used to perform operations on variables and values, such as arithmetic, comparisons, or logic. They enable developers to manipulate data and make decisions within the code.
Operators are very essential in programming because they allow to control the flow of execution,calculations and to help to apply condition in programm.we can not do any operation without the operators.
Example- if we want to calculate the sum of 3 number, then we need arithmetic operator.

```
 Let num1=10,num2=20,num3=20,sum;
sum=num1+num2+num3;   //using arithmetic operator for adding this num
console.log(sum);
```

Q.2-Describe the categorization of operator in javascript based on their functionality.Provide example for each category.
Ans-JavaScript operators are categorized following based on their functionality.
1-Arithmatic Operator- This operator is used for mathematic calculations and operations.
  They have plus(+) operator,minus(-) operator, multiplication(*) operator,divide(/) operator, modulus(%) operator etc.
Example- we want to calculate how many days in months so we use divide operator.

```
 Let daysInMonth=365/12;
console.log(daysInMonth);
```

2-Assigmnent Operator-This operator is used to assign the values to variables.
   There is different assignment operator like +=,-=,*=,/=.
Example-

```
let b=5;
b+=10;  // assigning 5 to b
```

3-Compaision Operators- This operator is used to compare the to variable's value.They have <,>,== operators.
Example-

```
let a=10;
let b=20 ;
console.log(b>a);
```

4-Logical Operator- This operator is used to perform logical operation and return boolean value true or false. They have AND (&&) operator, OR(||) operator and NOT (!) operator etc.
Example-

```
let a = true;
let b = false;
console.log(a && b);  // false (logical AND)
console.log(a || b);  // true (logical OR)
```

5-Bitwise Operator- This operator is used to perform the bitwise operations on numbers or integers.

They have bitwise AND, bitwise OR ,bitwise NOT.

Example-
```
let x = 5;  // (binary: 101)
let y = 3;  // (binary: 011)
console.log(x & y);  // 1 (bitwise AND)
```

6- Increment and Decreament Operator- This operator is used to increment or decreament the value of variables.

Example-
```
let a=10;
let b=15;
a++;  //increamt the value of a
b - -; //decreamnet value of b
```

Q.3- Differentiate between unary, binary and ternary operators in Javascript. Give Examples.

Ans-Operators are categorized based on number of operator using.

1-Unary Operator- In this operator, operator is only single operands.

Example-
```
let x = 10;
x++;  // x becomes 11 (post-increment)
--x;  // x becomes 10 (pre-decrement)
```

2- Binary Operator- This operator used two operands for any operations lke arithmetic operators(+,-,*,/,),comparision operators(<,>,==),logical operators(&&,||,!) etc.

Example-
```
let a = 10, b = 5;
let sum = a + b;  // 15
let multi = a * b;  // 50
console.log(sum,multi);
console.log(a > b);  // true
```

3- Ternary Operator- This operator is operates on three operands.

Syntax - condition? expr 1 :expr 2;

If the condition is true, it returns expr 1 otherwise, it returns expr 2.

Example-
```
let age = 5;
let admission = (age >= 4) ? "Yes" : "No";     // "Yes"
```

Q.4- Discuss the precedence and associativity of operator in JavaScript. Why is understanding of these concepts important?

Ans- Precedence of operators- Precedence determines the order in which operators are evaluated in expressions. Like multiplication (*) has a higher precedence than addition (+).

Example-
```
let result = 3 + 5 * 5;    // result = 28 (because * has higher precedence than +)
```

Operator Associativity- When two operators have the same precedence, associativity determines the direction in which the operators are evaluated.
Left-to-right associativity -means the operators are evaluated from the leftmost operator to the rightmost.
Right-to-left associativity- means the operators are evaluated from the rightmost to the leftmost.
Example- let result = 10 - 3 - 2; // result = 5 (left-to-right associativity for subtraction)

Importance Of precedence and Associativity-

Precedence and associativity of operators ensures that expressions are evaluated in the intended order, avoiding bugs or unexpected results.

To writing clean and efficient code so that any one can understand the code.