Name: Animesh Parab

Batch: T2- T21

Roll no:88

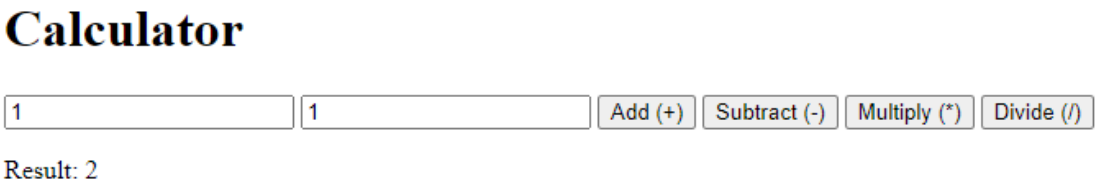
Aim: WAP in JS to study conditional Statements, Loops and Functions

Write a java script for simple calculation i.e. add, subtract, division, multiplication.

```
<!DOCTYPE html>
<html>
<head>
 <title>Calculator</title>
</head>
<body>
 <h1>Calculator</h1>

 <div class="calculator">
  <input type="number" id="num1" placeholder="Enter first number">
  <input type="number" id="num2" placeholder="Enter second number">

  <button id="add">Add (+)</button>
  <button id="subtract">Subtract (-)</button>
  <button id="multiply">Multiply (*)</button>
  <button id="divide">Divide (/)</button>

  <p id="result"></p>
 </div>
```

```
<script>
  function performOperation(operation) {
    const num1 = parseFloat(document.getElementById("num1").value);
    const num2 = parseFloat(document.getElementById("num2").value);
    let result;

    switch (operation) {
      case "add":
        result = num1 + num2;
        break;
      case "subtract":
        result = num1 - num2;
        break;
      case "multiply":
        result = num1 * num2;
        break;
      case "divide":
        if (num2 === 0) {
          document.getElementById("result").textContent = "Cannot divide by
zero";
          return;
        }
        result = num1 / num2;
        break;
    }

    document.getElementById("result").textContent = `Result: ${result}`;
  }
```

```
    document.getElementById("add").addEventListener("click", function() {
      performOperation("add");
    });

    document.getElementById("subtract").addEventListener("click", function() {
      performOperation("subtract");
    });

    document.getElementById("multiply").addEventListener("click", function() {
      performOperation("multiply");
    });

    document.getElementById("divide").addEventListener("click", function() {
      performOperation("divide");
    });
  </script>
</body>
</html>
```



Conclusion:
JavaScript's conditional statements, loops, and functions are integral components that empower developers to create dynamic and interactive web applications. By mastering these constructs, programmers can control program flow, efficiently manage repetitive tasks, and encapsulate reusable code. These foundational concepts form the basis of JavaScript programming and are essential for building sophisticated and responsive web experiences.