```
public class ScientificCalculatorCont {
    public Decimal num1 { get; set; }
    public Decimal num2 { get; set; }
    public String operation { get; set; }
    public Decimal result { get; set; }
    public String errorMessage { get; set; }

    public void calculate() {
        try {
            if (operation == 'Addition') {
                result = num1 + num2;
            } else if (operation == 'Subtraction') {
                result = num1 - num2;
            } else if (operation == 'Multiplication') {
                result = num1 * num2;
            } else if (operation == 'Division') {
                if (num2 == 0) {
                    errorMessage = 'Error: Division by zero is not allowed.';
                    return;
                }
                result = num1 / num2;
            } else if (operation == 'SquareRoot') {
                if (num1 < 0) {
                    errorMessage = 'Error: Cannot take the square root of a negative number.';
                    return;
                }
                result = Math.sqrt(num1);
            } else if (operation == 'Power') {
                // Use Double to handle power operation
                result = (Decimal) Math.pow((Double) num1, (Double) num2);
            } else if (operation == 'Sine') {
                result = Math.sin((Double) num1);
            } else if (operation == 'Cosine') {
                result = Math.cos((Double) num1);
            } else if (operation == 'Tangent') {
                result = Math.tan((Double) num1);
            } else if (operation == 'Log') {
                if (num1 <= 0) {
                    errorMessage = 'Error: Logarithm undefined for non-positive numbers.';
                    return;
                }
            result = Math.log((Double) num1);
        } else {
            errorMessage = 'Invalid operation';
        }
    } catch (Exception e) {
        errorMessage = 'Error: ' + e.getMessage();
    }
  }
}
```

```
<apex:page controller="ScientificCalculatorCont">
  <h2>Scientific Calculator</h2>
  <apex:form >
    <apex:pageBlock title="Enter Values">
      <apex:pageBlockSection >
        <apex:inputText value="{!num1}" label="Enter First Number" />
        <apex:inputText value="{!num2}" label="Enter Second Number (if applicable)" />
        <apex:selectList value="{!operation}" size="1">
          <apex:selectOption itemLabel="Select Operation" itemValue="" />
          <apex:selectOption itemLabel="Addition" itemValue="Addition" />
          <apex:selectOption itemLabel="Subtraction" itemValue="Subtraction" />
          <apex:selectOption itemLabel="Multiplication" itemValue="Multiplication" />
          <apex:selectOption itemLabel="Division" itemValue="Division" />
          <apex:selectOption itemLabel="Square Root" itemValue="SquareRoot" />
          <apex:selectOption itemLabel="Power" itemValue="Power" />
          <apex:selectOption itemLabel="Sine" itemValue="Sine" />
          <apex:selectOption itemLabel="Cosine" itemValue="Cosine" />
          <apex:selectOption itemLabel="Tangent" itemValue="Tangent" />
          <apex:selectOption itemLabel="Logarithm" itemValue="Log" />
        </apex:selectList>
      </apex:pageBlockSection>

      <apex:pageBlockSection >
        <apex:commandButton value="Calculate" action="{!calculate}"
rerender="resultSection"/>
      </apex:pageBlockSection>

      <apex:pageBlockSection id="resultSection">
        <apex:outputPanel rendered="{!NOT(ISBLANK(errorMessage))}">
          <apex:outputText value="{!errorMessage}" style="color: red;" />
        </apex:outputPanel>

        <apex:outputPanel rendered="{!NOT(ISBLANK(result))}">
          <apex:outputLabel value="Result: " />
          <apex:outputText value="{!result}" />
        </apex:outputPanel>
      </apex:pageBlockSection>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

```
// Create an instance of the ScientificCalculatorCont class
ScientificCalculatorCont calculator = new ScientificCalculatorCont();

// Test for Addition
calculator.num1 = 10;
calculator.num2 = 5;
calculator.operation = 'Addition';
calculator.calculate();
System.debug('Result of Addition: ' + calculator.result);  // Expected: 15

// Test for Subtraction
calculator.operation = 'Subtraction';
calculator.calculate();
System.debug('Result of Subtraction: ' + calculator.result);  // Expected: 5

// Test for Multiplication
calculator.operation = 'Multiplication';
calculator.calculate();
System.debug('Result of Multiplication: ' + calculator.result);  // Expected: 50

// Test for Division
calculator.operation = 'Division';
calculator.calculate();
System.debug('Result of Division: ' + calculator.result);  // Expected: 2

// Test for Square Root
calculator.num1 = 16;
calculator.operation = 'SquareRoot';
calculator.calculate();
System.debug('Result of Square Root: ' + calculator.result);  // Expected: 4
```

```
// Test for Power
calculator.num1 = 2;
calculator.num2 = 3;
calculator.operation = 'Power';
calculator.calculate();
System.debug('Result of Power: ' + calculator.result);  // Expected: 8

// Test for Sine (in radians)
calculator.num1 = 3.14159 / 2; // Pi/2
calculator.operation = 'Sine';
calculator.calculate();
System.debug('Result of Sine: ' + calculator.result);  // Expected: 1

// Test for Cosine (in radians)
calculator.operation = 'Cosine';
calculator.calculate();
System.debug('Result of Cosine: ' + calculator.result);  // Expected: 0

// Test for Tangent (in radians)
calculator.operation = 'Tangent';
calculator.calculate();
System.debug('Result of Tangent: ' + calculator.result);  // Expected: Undefined or some value based on angle

// Test for Logarithm
calculator.num1 = 10;
calculator.operation = 'Log';
calculator.calculate();
System.debug('Result of Logarithm: ' + calculator.result);  // Expected: log(10) ~ 2.302585
```