```
#include <iostream>
#include <cmath>

// Function to calculate sin(x) using the expansion formula: sin(x) = ∑ (-1)^n / ((2n + 1)!) * x^(2n + 1)
double calculateSin(double x, double precision) {
    double result = x;  // Set the initial result to the value of x
    double term = x;    // Set the initial term to the value of x
    int n = 1;          // Initialize the counter

    while (abs(term) >= precision) {
        double numerator = -term * x * x;           // Calculate the numerator of the next term: -term * x^2
        double denominator = (2 * n) * (2 * n + 1);  // Calculate the denominator of the next term: (2n) * (2n +
1)

        term = numerator / denominator;             // Calculate the next term
        result += term;                             // Add the next term to the result
        n++;                                        // Increment the counter
    }

    return result;
}

int main() {
    double x;
    double precision = 0.0001;
```

```
    cout << "Enter the value of x: ";
    cin >> x;

    double result = calculateSin(x, precision);

    cout << "sin(" << x << ") = " << result << std::endl;

    return 0;
}
```

EXPLANATION

The calculateSin function takes in the value of x and the desired precision as parameters and returns the calculated value of sin(x).

Inside the function, we initialize result and term to the initial value of x. The variable result will store the accumulated sum of terms, and term represents each term in the series.

We start a while loop that continues until the absolute value of term becomes smaller than the given precision.

Inside the loop, we calculate the numerator and denominator of the next term using the formulas -term * x * x and (2 * n) * (2 * n + 1) respectively. The numerator involves the current term and x^2, while the denominator represents the factorial term in the expansion formula.

We update term by dividing the numerator by the denominator, and then add term to the result.

Finally, we increment the counter n to move to the next term in the series.

Once the loop ends, we return the calculated result.

In the main function, we prompt the user to enter the value of x and store it in the variable x.

We call the calculateSin function with the given x and precision values, and store the result in the variable result.

Finally, we display the result as sin(x) = result on the console.

The code provides a detailed explanation of the implementation, along with the expansion formula mentioned in comments, to help understand the calculation of sin(x) using the provided iterative algorithm.