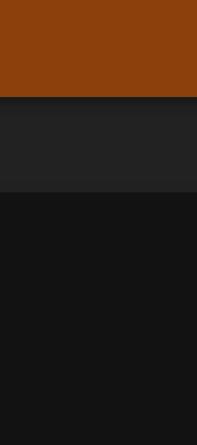


↑ 6.28 Functions with 2D array parameters



Students:
Section 6.29 is a part of 1 assignment: CSC108 CH06.10-6.29 P6B

↳ Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Includes: PA
Due: 05/06/2025, 11:59 PM EDT

6.29 Functions with array parameters: Common errors

For arrays other than C strings, a common error is defining a function with an array parameter without a second parameter indicating the number of array elements. Without a parameter indicating the number of array elements, the function will only work for one specific array size. A programmer may call the function with a larger or smaller array, which may lead to incorrect results.

PARTICIPATION ACTIVITY | 6.29.1: Common error: Functions with arrays without a parameter indicating the number of array elements.

Start 2x speed

```
float GetAvgScore(float scoreVals[]) {
    int i;
    float scoreSum = 0;

    for (i = 0; i < 5; ++i) {
        scoreSum = scoreSum + scoreVals[i];
    }

    return scoreSum / 5.0;
}

scoreSum = 0.0 + 90.0 + 95.5 + 97.0 + 80.0 + 85.0
= 447.5
```

```
hwAvg = GetAvgScore(hwScores); 83.2 ✘ Average of all 7 scores in hwScores
                                         should be 81.214

examAvg = GetAvgScore(examScores); 89.5 ✘ Average incorrectly includes values
                                         outside the array bounds
```

Captions ▾

Feedback?

PARTICIPATION ACTIVITY | 6.29.2: Function with array missing parameter indicating number of array elements.

1) The GetMinScore() function has a common error of not including a parameter to indicate the number of array elements. Given the arrays preGames, seasonGames, and playoffGames, which function call does not return the minimum value?

```
int GetMinScore(int scoreVals[]) {
    int i;
    int minScore;

    minScore = scoreVals[0];
    for (i = 0; i < 5; ++i) {
        if (scoreVals[i] < minScore) {
            minScore = scoreVals[i];
        }
    }

    return minScore;
}

int preGames[] = { 80, 45, 86, 85, 64, 99, 98 };
int seasonGames[] = { 92, 90, 84, 82, 98 };
int playoffGames[] = { 85, 83, 75, 90, 70, 68, 69 };

● GetMinScore(preGames);
● GetMinScore(seasonGames);
● GetMinScore(playoffGames);
```

2) Which function definition correctly supports arrays of any size?

```
int GetMaxScoreA(int scoreVals[]) {
    int i;
    int maxScore;

    maxScore = scoreVals[0];
    for (i = 0; i < sizeof(scoreVals); ++i) {
        if (scoreVals[i] > maxScore) {
            maxScore = scoreVals[i];
        }
    }

    return maxScore;
}
```

```
int GetMaxScoreB(int scoreVals[], int numScores) {
    int i;
    int maxScore;

    maxScore = scoreVals[0];
    for (i = 0; i < numScores; ++i) {
        if (scoreVals[i] > maxScore) {
            maxScore = scoreVals[i];
        }
    }

    return maxScore;
}
```

● GetMaxScoreA()
● GetMaxScoreB()

Feedback?

Another common error is modifying array elements in a function that should not modify the array. Arrays are automatically passed to functions using a pointer to the array. So, a function that modifies an array parameter will modify the array argument passed to the function, and not a local copy.

The FindMaxAbsValueIncorrect() function incorrectly modifies the inputVals array to find the array element with the largest absolute value.

Figure 6.29.1: FindMaxAbsValueIncorrect() incorrectly modifies inputVals array to find element with the largest absolute value.

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

using namespace std;

float FindMaxAbsValueIncorrect(float inputVals[], int numVals) {
    int i;
    float maxAbsVal;

    // Incorrectly updates inputVals to calculate absolute value
    // of array elements
    for (i = 0; i < numVals; ++i) {
        inputVals[i] = fabs(inputVals[i]);
    }

    maxAbsVal = inputVals[0];
    for (i = 0; i < numVals; ++i) {
        if (inputVals[i] > maxAbsVal) {
            maxAbsVal = inputVals[i];
        }
    }

    return maxAbsVal;
}

int main() {
    const int NUM_VALUES = 5;
    // Array of changes in temperatures
    float tempChanges[NUM_VALUES] = {10.0, 0.5, -5.1, -11.2, 3.0};
    float maxAbsChange;
    int i;

    // Print array before function call
    cout << "tempChanges array before function call: ";
    for(i = 0; i < NUM_VALUES; ++i) {
        cout << tempChanges[i] << " ";
    }
    cout << endl;

    // Find the largest temperature change, and print result.
    maxAbsChange = FindMaxAbsValueIncorrect(tempChanges, NUM_VALUES);
    cout << "Max absolute temperature change: " << maxAbsChange << endl;

    // Print array after function call
    cout << "tempChanges array after function call: ";
    for(i = 0; i < NUM_VALUES; ++i) {
        cout << tempChanges[i] << " ";
    }
    cout << endl;
}

tempChanges array before function call: 10 0.5 -5.1 -11.2 3
Max absolute temperature change: 11.2
tempChanges array after function call: 10 0.5 5.1 11.2 3
```

Feedback?

The FindMaxAbsValue() function performs the same computation but avoids modifying the inputVals array. The FindMaxAbsValue() also defines the inputVals as const, which prevents the function from modifying the parameter. For arrays that should not be modified in a function, good practice is to define the array parameter as const.

Figure 6.29.2: FindMaxAbsValue() computes the largest absolute value without modifying the array.

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

using namespace std;

float FindMaxAbsValue(const float inputVals[], int numVals) {
    int i;
    float maxAbsVal;
    float inputAbsVal;

    maxAbsVal = fabs(inputVals[0]);
    for (i = 0; i < numVals; ++i) {
        inputAbsVal = fabs(inputVals[i]);
        if (inputAbsVal > maxAbsVal) {
            maxAbsVal = inputAbsVal;
        }
    }

    return maxAbsVal;
}

int main() {
    const int NUM_VALUES = 5;
    // Array of changes in temperatures
    float tempChanges[NUM_VALUES] = {10.0, 0.5, -5.1, -11.2, 3.0};
    float maxAbsChange;
    int i;

    // Print array before function call
    cout << "tempChanges array before function call: ";
    for(i = 0; i < NUM_VALUES; ++i) {
        cout << tempChanges[i] << " ";
    }
    cout << endl;

    // Find the largest temperature change, and print result.
    maxAbsChange = FindMaxAbsValue(tempChanges, NUM_VALUES);
    cout << "Max absolute temperature change: " << maxAbsChange << endl;

    // Print array after function call
    cout << "tempChanges array after function call: ";
    for(i = 0; i < NUM_VALUES; ++i) {
        cout << tempChanges[i] << " ";
    }
    cout << endl;
}

tempChanges array before function call: 10 0.5 -5.1 -11.2 3
Max absolute temperature change: 11.2
tempChanges array after function call: 10 0.5 -5.1 -11.2 3
```

Feedback?

PARTICIPATION ACTIVITY | 6.29.3: Functions array parameters.

Based on the function names, should the following functions allow the array parameters to be modified?

1) void PrintReverse(int inputVals[], int numVals)

- Yes
- No

2) void SortArrayAscending(int inputVals[], int numVals)

- Yes
- No

3) int FindIndexSmallest(int inputVals[], int numVals)

- Yes
- No

Feedback?

How was this section? Provide section feedback

Activity summary for assignment: CSC108 CH06.10-6.29 P6B

Due: 05/06/2025, 11:59 PM EDT

0 / 91 points

↳ Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Completion details ▾

↓ 6.30 C++ example: Salary calculation with functions