6.26 Memory leaks

Students:
Section 6.27 is a part of 2 assignments: CSC108 CH06.10-6.27 C6B

Please browse to this assignment through BlackboardLearn so zybooks knows where to send your activity. Learn more

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

6.27 Functions with array parameters

Functions commonly have array parameters. The following program uses a function to calculate the average value of the elements within an array of test scores.

Figure 6.27.1: Array parameters in an average test score calculation program.

```
#include <iostream>
using namespace std;

double CalculateAverage(double scoreVals[], int numVals) {
  int index;
  double scoreSum = 0;

  for (index = 0; index < numVals; ++index) {
    scoreSum = scoreSum + scoreVals[index];
  }

  return scoreSum / numVals;

}

int main() {
  const int NUM_SCORES = 5;   // Array size
  double testScores[NUM_SCORES];  // User test scores
  int index;
  double averageScore;

  // Prompt user to enter test scores
  cout << "Enter " << NUM_SCORES << " test scores:" << endl;
  for (index = 0; index < NUM_SCORES; ++index) {
    cout << "Test score: ";
    cin >> testScores[index];

  cout << endl;

  // Call function to calculate average
  averageScore = CalculateAverage(testScores, NUM_SCORES);
  cout << "Average test score: "
       << averageScore << endl;

  return 0;
}
```

```
Enter 5 test scores:
Test score: 95.5
Test score: 92.5
Test score: 97.5
Test score: 97.0

Average test score: 95.75
```

Feedback?

The parameter definition uses [] to indicate an array parameter. The function call's argument does not use []. The compiler automatically passes array arguments by passing a pointer to the array, rather than making a copy like for other parameter types, in part to avoid the inefficiency of copying large arrays.

Functions with array parameters, other than C string parameters (discussed elsewhere), need a second parameter indicating the number of elements in the array. Ex: numVals indicates the number of elements within the scoreVals array.

PARTICIPATION ACTIVITY 6.27.1: Functions with array parameters.

1) Passing an array creates a copy of that array within the function.
- True
- False

2) An array is automatically passed by pointer.
- True
- False

Feedback?

PARTICIPATION ACTIVITY 6.27.2: Functions with arrays.

1) Complete the FindMin function definition to have an array parameter of type double named arrayVals and second parameter of type int numVals indicating the number of array elements.

```
double FindMin(                                  ) {
  // ...
}
```

Check Show answer

2) Complete the PrintVals function to print each array element on a separate line.

```
void PrintVals(int arrayVals[], int numElements) {
  int i;

  for(i = 0; i                          ; ++i) {
    cout << arrayVals[i] << endl;
}
```

Check Show answer

3) sellingPrices is an array of type double with 10 elements. Complete the statement to find the lowest price by calling the GetLowestPrice function defined below.

```
GetLowestPrice(double itemPrices[], int numItems);

lowestPrice = GetLowestPrice(                    );
```

Check Show answer

Feedback?

Because an array is passed to a function by passing a pointer to the array, a function can modify the elements of an array argument.

PARTICIPATION ACTIVITY 6.27.3: Functions can modify elements of an array argument.

Start 2x speed

```
void AdjustScores(double scoreVals[], int numVals) {
  int i;

  for (i = 0; i < numVals; ++i) {
    scoreVals[i] = scoreVals[i] + scoreAdj;
  }

int main() {
  const int NUM_SCORES = 4;   // Array size
  double testScores[NUM_SCORES];  // Test scores
  // ...

  AdjustScores(testScores, NUM_SCORES, 2.5);

  return 0;
}
```

Captions

Feedback?

The keyword **const** can be prepended to a function's array parameter to prevent the function from modifying the parameter. In the following program, the AdjustScores function modifies the array so defines a normal array parameter. The PrintScores and CalculateAverage functions do not modify the array so define a const array parameter.

Figure 6.27.2: Normal and const array parameters in test score adjustment and averaging program.

```
#include <iostream>
using namespace std;

void AdjustScores(double scoreVals[], int numVals,
                  double scoreAdj) {
  int i;

  for (i = 0; i < numVals; ++i) {
    scoreVals[i] = scoreVals[i] + scoreAdj;
  }
}

void PrintScores(const double scoreVals[], int numVals) {
  int i;

  for (i = 0; i < numVals; ++i) {
    cout << " " << scoreVals[i];
  }

  cout << endl;

}

double CalculateAverage(const double scoreVals[], int numVals) {
  int i;
  double scoreSum = 0;

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

  return scoreSum / numVals;

}

int main() {
  const int NUM_SCORES = 4;   // Array size
  double testScores[NUM_SCORES];  // User test scores
  int i;
  double averageScore;

  // Prompt user to enter test scores
  cout << "Enter " << NUM_SCORES << " test scores:" << endl;
  for (i = 0; i < NUM_SCORES; ++i) {
    cout << "Test score: ";
    cin >> testScores[i];

  cout << endl;

  // Print original scores
  cout << "Original test scores: ";
  PrintScores(testScores, NUM_SCORES);

  AdjustScores(testScores, NUM_SCORES, 2.5);

  cout << "Adjusted test scores: ";
  PrintScores(testScores, NUM_SCORES);

  // Call function to calculate average
  averageScore = CalculateAverage(testScores, NUM_SCORES);
  cout << "Average adjusted test score: "
       << averageScore << endl;

  return 0;
```

```
Enter 4 test scores:
Test score: 95.0
Test score: 97.5
Test score: 99.0
Test score: 94.5

Original test scores:  92 95.5 97 92.5
Adjusted test scores:  92.5 95.75 99 94.5
Average adjusted test score: 95.75
```

Feedback?

PARTICIPATION ACTIVITY 6.27.4: Const array function parameters.

Can the following functions modify the array parameter?

1) void PrintArray(double userRuns[], int numElements)
- Yes
- No

2) int GetHighScore(int userScores[], const int numHighScores)
- Yes
- No

3) int FindIndex(const int searchVals[], int numVal, int key)
- Yes
- No

Feedback?

CHALLENGE ACTIVITY 6.27.1: Enter the output of the function with array parameters.

zybooks.zyante.com/cpp/7

Start

Type the program's output

```
#include <iostream>
using namespace std;

void PrintSortedArray(const int scoreVals[], int numVals) {
  int i;

  for (i = 0; i < numVals; ++i) {
    cout << scoreVals[i] << endl;
  }
}

int main() {
  const int NUM_SCORES = 4;
  int quizScores[NUM_SCORES];
  int i;

  for (i = 0; i < NUM_SCORES; ++i) {
    cin >> quizScores[i];

  PrintSortedArray(quizScores, NUM_SCORES);

  return 0;
```

Input
3
0
8
1

Output
3
0
8
1

1 2 3 4

Check Next

Feedback?

CHALLENGE ACTIVITY 6.27.2: Modify an array parameter. Full screen

zybooks.zyante.com/cpp/7

Organize the lines of code to define a function SwapArrayEnds() that swaps the first and last elements of the function's array parameter.

Ex: If the input is 10 20 30 40, then the output is:

40 20 30 10

Note: Not all lines of code on the left will be used in the final solution.

How to use this tool

Unused

```
int temp;value;
inputArray[sizeOfArray - 1] = temp;value;
temp;value = inputArray[0];
void SwapArrayEnds(int inputArray[], const int sizeOfArray) {
void SwapArrayEnds(int inputArray[], int sizeOfArray) {
void SwapArrayEnds(const int inputArray[], int sizeOfArray) {

inputArray[0] = inputArray[sizeOfArray - 1];
```

Check

main.cpp Load default template...

```
#include <iostream>
using namespace std;

int main() {
  const int SORT_ARR_SIZE = 4;
  int sortArray[SORT_ARR_SIZE];
  int i;
  int userNum;

  for (i = 0; i < SORT_ARR_SIZE; ++i) {
    cin >> userNum;
    sortArray[i] = userNum;
  }

  SwapArrayEnds(sortArray, SORT_ARR_SIZE);

  for (i = 0; i < SORT_ARR_SIZE; ++i) {
    cout << sortArray[i] << " ";
```

Feedback?

CHALLENGE ACTIVITY 6.27.3: Functions with array parameters.

zybooks.zyante.com/cpp/7

Start

Four integers are read from input and stored into the array myArray. Write a function ModifyFirstValue() that takes an integer array parameter and replaces the first element with -1.

Ex: If the input is 125 40 140 75, then the output is:

Input: 125 40 140 75
Output: -1 40 140 75

```
#include <iostream>
using namespace std;

/* Your code goes here */

void PrintArr(const int arr[], int numVals) {
  int i;

  for (i = 0; i < numVals; ++i) {
    cout << arr[i] << " ";
  }

  cout << endl;

}

int main() {
  const int NUM_ELEMENTS = 4;
  int myArray[NUM_ELEMENTS];
```

1 2 3 4

Check Next level

Feedback?

How was this section? Provide section feedback

Activity summary for assignment: CSC108 CH06.10-6.27 C6B
Due: 05/06/2025, 11:59 PM EDT

0 / 30 points

Please browse to this assignment through BlackboardLearn so zybooks knows where to send your activity. Learn more

Completion details