Passing Whole Arrays to Functions

Larry Caretto
Computer Science 106

Computing in Engineering and Science

April 27, 2006

California State University Northridge

Outline

- Review introduction to arrays
- Review writing code with arrays and for loops
- Review data processing with arrays
- Arrays and functions
  - Passing array elements to functions
  - Passing whole arrays to functions
  - Writing functions that have whole arrays as arguments

Representing Data

Run	Data	Math	C++
0	12.3	x_0	x[0]
1	14.4	x_1	x[1]
2	11.8	x_2	x[2]
3	12.5	x_3	x[3]
4	13.2	x_4	x[4]
5	14.1	x_5	x[5]

- C++ array, x[k] used to represent data for which x_k is used in mathe-matical notation

Using Arrays

- Declare arrays in typical way, but add maximum elements, e.g. int v[100];
- Refer to arrays as to any other variable using subscript v[3] or v[k]
  - Must assign value to k before using it as variable subscript
  - Major tool in arrays is using variable subscript that is for loop index

```
const int N = 200; double a[N];
for ( int j = 0; j < N; j++) a[j] = 0;
```


Maximum Array Subscript

- Array subscripts start at zero
- A declaration double y[N] declares a y array with N elements numbered from y[0] to y[N-1]
  N must be a const
- For loop to handle all elements is

```
for ( int k = 0; k < N; k++ )
```

- C++ does not check to see if an array subscript is in bounds -- an incorrect subscript could affect some other memory location

General Array Processing

- To process each element in an array with N elements, starting with the initial element, use a for loop with index k
  - k starts at zero
  - The continuation condition, k < N, will process elements 0, 1, 2, … , N-1
  - Increment k by 1
- `for ( int k = 0; k < N; k++)`
- Will process elements 0 to N – 1 of array regardless of array size

Passing Arrays to Functions

- We can pass an array element to a function as we pass any variable
- y = pow(x[k], 3);
- Here the pow function returns the cube of element k of the x array
- This is no different from passing a single variable to a function
- We can also pass whole arrays, like x, to functions: getAverage(x, first, last)

California State University Northridge 7

getAverage

- Computes the average of elements of the x array from x[first] to x[last] (inclusive)
- Header: double getAverage (double x[], int first, int last)
- Prototypes:
  - double getAverage (double x[], int first, int last);
  - double getAverage (double [], int, int);
- Note use of [] to specify an array as a function argument

California State University Northridge 8

getAverage

```
double getAverage ( double
 x[],          int first,
 int last )
{
   double sum = 0;
   for ( int i = first;
               i <= last;
 i++ )
          sum += x[i];
```

California State University Northridge 9

Use of getAverage

- double x[22], power[50], density[30];
- // code to get input data on x and power
- double mean = getAverage(x, 0, 10);
- double average = getAverage(power, 12, 24);
- How would you compute the average of all elements of the density array?

```
getAverage( density, 0, 29);
```

California State University Northridge 10

Standard Deviation

- Measure of spread around mean

$$s = \sqrt{\frac{\sum_{i=0}^{N-1}(x_i - \bar{x})^2}{N-1}} = \sqrt{\frac{\left(\sum_{i=0}^{N-1} x_i^2\right) - N(\bar{x})^2}{N-1}} = \sqrt{\frac{\left(\sum_{i=0}^{N-1} x_i^2\right) - \frac{1}{N}\left(\sum_{i=0}^{N-1} x_i\right)^2}{N-1}}$$

- First term is definition; others are computational forms
- How would we write a function to compute s for all the elements in an N-element array?

California State University Northridge 11

Write Function getStdDev

- Want to find s for array, x, with size N

$$s = \sqrt{\frac{\left(\sum_{i=0}^{N-1} x_i^2\right) - \frac{1}{N}\left(\sum_{i=0}^{N-1} x_i\right)^2}{N-1}}$$

- What is function header?

```
double getStdDev( double x[], int N)
```

- What is prototype?

```
double getStdDev( double x[], int N);
```

- What call gives s for double power[100]?

```
s = getStdDev( power, 100);
```

California State University Northridge 12

getStdDev

```
double getStdDev(double x[],int N)
{
    double sum = 0, sum2 = 0;
    for ( int k = 0; k < N; k++ )
    {
        sum += x[k];
        sum2 += x[k] * x[k];
    }
    return sqrt(
        ( sum2 - sum * sum / N )
        / ( N - 1 ) );
}
```


Arrays Passed by Reference

```
void setArray( double x[], int N,
               double value )
{
    for ( int k = 0, k < N; k++ )
        x[k] = value;
}
```

- A call, setArray(c, M) would zero the elements 0 to M – 1 of the c array
- For arrays, pass by reference occurs by default without the need for an &

Array Function Exercise

- Write a function that finds the minimum value of a type double array
- What information do you have to pass to the function?
  - The array name (double) and the number of elements in the array (int)
- How would you return the minimum to the calling function
  - In the name of the function

Array Function Exercise II

- Write a function that finds the minimum value of a type double array (continued)
- Write the function header

```
double getMin( double x[], int N )
```

- Write a statement that finds the minimum value of power: `double power[250]`

```
double y = getMin( power, 250 )
```

- What is `getMin( power, 100 )`?
  - The minimum value of elements 0 to 99 of an array named power

Array Function Exercise III

- Write a function that finds the minimum value of a type double array (continued)
- Write the prototype for getMin
- Two possible prototypes

```
double getMin( double x[], int N );
double getMin( double [], int  );
```

- Write the complete code for the function

Array Function Exercise IV

- The code for a function that finds the minimum value of a type double array

```
double getMin( double x[], int N )
{
   double min = x[0];
   for ( int k = 1; k < N; k++ )
       if ( x[k] < min )
           min = x[k];

   return min;
}
```

Data Input Function for Array

- If we know meaning of array parameters, we do not to know code
- bool getInputOK (double x[], int& n, int max) is used to read data into an array x and return an value of true if there are no errors (false if there are errors)
- Function returns the number of elements in the array, n
- Error if attempt to read n > max

Data Input Function for Array II

- Prototype for function

```
bool getInputOK ( double x[],
                  int&, m  int max );
```

- Use of function in code

```
const int MAX = 1000;
double y[MAX]
if ( !getInputOK( y, n, MAX ) ) {
    cout << "Program halt: "
         << "Data input error";
    return EXIT_FAILURE;
}
```


Array Input Function Code

```
bool getInputOK( double x[]
                 int& n, int max )
{
    // open file; check status
    ifstream inFile
             ("inputFile.dat" );
    if ( !inFile.good() )  {
       cout << "Could not "
            << "open input "
            << "data file.\n";
   return false;
}
```


Array Input Function Code II

```
      // Get input data
    n = 0;
    do {
        double xin;
        inFile >> xin;
        if ( inFile.fail() )
           break;
        x[n] = xin;
        n++;
    } while ( n < max );
// can exit if last data read
// or if >= max data items
```


Array Input Function Code III

```
 // Check for more data on file
   if ( inFile.good() ) {
      inFile >> xin;
      if ( !inFile.fail() ) {
         cout << "Array size "
              << "less than "
              << "number of "
              << "data on file";
         return false;
   }
   return true; // No errors
}
```