

Lab # 15: Arrays and Strings – Part 1

EC-102 – Computer Systems and Programming

Usman Ayub Sheikh

School of Mechanical and Manufacturing Engineering (SMME),
National University of Sciences and Technology (NUST)

Monday 14th December, 2015

Outline

1 Arrays

- Why?
- What?
- How?
 - Defining Array
 - Array Elements
 - Accessing Array Elements

2 Solved Examples

- Averaging Array Elements
- Initializing Arrays
- Multi-dimensional Arrays
 - Defining Multi-dimensional Array
 - Accessing Multi-dimensional Array Elements
 - Initializing Multi-dimensional Arrays

3 Exercise

- Exercise 1
- Exercise 2

Why Array?

- In everyday life, we commonly **group** similar objects into units
 - A crate of peas, and
 - A dozen of eggs
- In computer programs, we also need to group together data items **of the same type**
- The most basic mechanism that accomplishes this in C++ is the **array**
- The data items grouped in an array can be simple types such `int`, `float` or they can be user-defined types such as structures

What is an Array?

- Like structures, arrays also group a number of items into a larger unit
- But while structures usually group items of different types, an array groups items of the same type
- Whereas items in a structure are accessed by name, those in an array **are accessed by an index number**

A Simple Program Using Array

```
1 // gets four ages from user and displays them
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int age[4];
8     for(int j = 0; j < 4; j++)
9     {
10         cout << "Enter an age: ";
11         cin >> age[j];
12     }
13
14     for(int j = 0; j < 4; j++)
15     {
16         cout << "You entered: " << age[j] << endl;
17     }
18     return 0;
19 }
```

A Simple Program Using Array

Defining Array:

- An array must be defined before it can be used
- An array definition specifies **a variable type** and **a name**
- Unlike variable definitions, it includes another feature: **a size**

```
int age[4];
```

Array Elements:

- The items in an array
- All the elements in an array are of the **same type**
- The first element is numbered 0

Accessing Array Elements:

- Name of the array followed by brackets delimiting a variable or a constant known as the **array index**
- `age[j]` refers to the j^{th} element of the array `age`

Averaging Array Elements

```
1 // averages a weeks's sales
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     const int SIZE = 6;
8     double sales[SIZE];
9
10    cout << "Enter widget sales for 6 days:\n";
11    for (int j = 0; j < SIZE; j++)
12        cin >> sales[j];
13
14    double total = 0;
15    for (int j = 0; j < SIZE; j++)
16        total += sales[j];
17
18    cout << "Average: " << total / SIZE << endl;
19    return 0;
20 }
```

Averaging Array Elements

- Using a variable (SIZE) makes it easier to change the array size
- Only one program line needs to be changed to change the array size, loop limits and anywhere else the array size appears
- Using all upper-case name reminds us that the variable cannot be modified in the program

Initializing Arrays

```
1 // shows days from start of the year to date specified
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int month, day, total_days;
7     int days_per_month[12] = {31, 28, 31, 30, 31, 30, 31, 31,
8     30, 31, 30, 31};
9
10    cout << "Enter month (1 to 12): ";
11    cin >> month;
12    cout << "Enter day (1 to 31): ";
13    cin >> day;
14    total_days = day;
15    for (int j = 0; j < month - 1; j++)
16        total_days += days_per_month[j];
17    cout << "Total days from start of the year is: " <<
18    total_days << endl;
19    return 0;
20 }
```

Initializing Arrays

- The values to which an array is initialized **are surrounded by braces** and **are separated by commas**

```
int days_per_month[12] = {31, 28, 31, 30, 31, 30, 31, 31,  
                          30, 31, 30, 31};
```

- What happens if you do use an explicit array size but it does not agree with the number of initializers?
 - If there are too few initializers, **the missing elements will be set to 0**
- If there are too many, **an error is signaled**

```
int days_per_month[12] = {31, 28, 31, 30, 31, 30, 31};  
In this case, days_per_month[7] to [11] will all be set to 0
```

```
int days_per_month[12] = {31, 28, 31, 30, 31, 30, 31, 31,  
                          30, 31, 30, 31, 30, 31, 30, 31};
```

Multi-dimensional Arrays

```
1 // displays sales chart using 2-d array
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 const int DISTRICTS = 4; // array dimensions
7 const int MONTHS = 3;
8
9 int main()
10 {
11     int d, m;
12     double sales[DISTRICTS][MONTHS]; // 2-d array definition
13     cout << endl;
14     for (d = 0; d < DISTRICTS; d++)
15         for (m = 0; m < MONTHS; m++)
16             {
17                 cout << "Enter sales for district " << d + 1;
18                 cout << ", month " << m + 1 << ": ";
19                 cin >> sales[d][m];
20             }
```

Multi-dimensional Arrays

```
21 cout << "\n\n";
22 cout << "                                Month\n";
23 cout << "                                1           2           3";
24 for (d = 0; d < DISTRICTS; d++)
25 {
26     cout << "\nDistrict " << d + 1;
27     for (m = 0; m < MONTHS; m++)
28     {
29         cout << setw(10) << sales[d][m];
30     }
31 }
32 cout << endl;
33 return 0;
34 }
```

Multi-dimensional Arrays

Defining Multi-dimensional Arrays:

- It is an array of DISTRICTS elements, each of which is an array of MONTHS elements

```
double sales[DISTRICTS][MONTHS];
```

- There can be arrays of more than two dimensions e.g. a three dimensional array is an array of arrays of arrays

Accessing Array Elements:

- Each index has its own set of brackets

```
sales[d][m]
```

- Commas are not used

Multi-dimensional Arrays

Initializing Multi-dimensional Arrays:

The initializing values for each sub-array are enclosed in braces and separated by commas

```
1  double sales[DISTRICTS][MONTHS] = {  
2      {1432.07, 234.50, 654.01},  
3      {322.00, 13838.32, 17589.88},  
4      {9328.34, 934.00, 4492.30},  
5      {12838.29, 2332.63, 32.93}  
6  };
```

Exercise 1

Write a program that:

- Declares an array containing ten elements from 1 to 10
- Multiplies the number entered by the user with this array, and
- Displays the output

Exercise 2

Write a program that:

- Asks the user to enter 5 numbers
- Writes those numbers to an array
- Arranges the elements of the array in ascending order, and
- Displays the array