CMP 1001

Introduction to Programming

Part 6: Arrays in C++

by

Assist. Prof. M. Şükrü Kuran

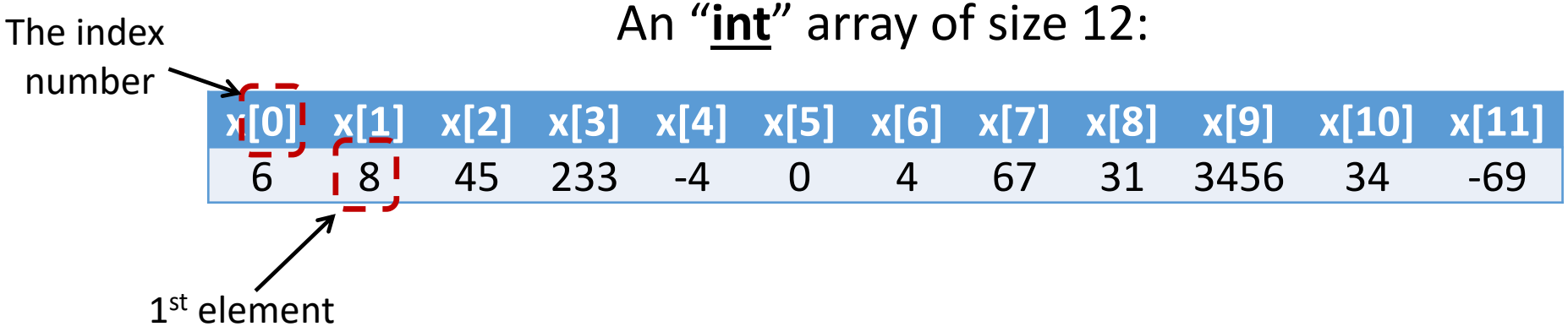
CMP 1001

Section 8: Arrays

ARRAYS

Keeping a Group of Variables

An "**int**" array of size 12:



x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]	x[10]	x[11]
6	8	45	233	-4	0	4	67	31	3456	34	-69

NOTE: The first element of each array is **ALWAYS** called the 0th element.

NOTE: The indices can only be **>= 0**

ARRAYS

Declaration of Arrays

Declaring an Integer Array of size 12:

```
int arrayName[12];
```

"**Arrays**" are **objects** (non-primitive types) that occupy space in memory.

NOTE: The **LAST** element in an array of size "n" is the element with index "n-1"

ARRAYS

Use of Arrays

After declaring the array, we can use each element of an array like a variable:

```
arrayName[3] = 2;
arrayName[4] = arrayName [3] * 5;
arrayName[5]++;
```

Example:
Write a program that reads an integer N, then reads that many integers and calculate the average of these numbers.

We can also initialize arrays like other primitive data types.

```
int n[] = {10, 40, 3, -2, 99};
```

ARRAYS

Example

We can get the **LENGTH** of an array by using the sizeof() function.

```
int d[23];
int size = sizeof(d)/sizeof(d[0]);
std::cout << "The length of array d is " << size);
```

Example:

Write a program that reads an integer N, and then reads N integer elements from the user. You want to check whether this sequence of numbers is symmetric or not.
(i.e., 1 2 3 4 5 5 4 3 2 1 is symmetric, 1 2 3 4 5 6 4 3 2 1 is NOT symmetric)

ENHANCED FOR STATEMENT

```
int size = sizeof(array)/sizeof(array[0]);

for (int counter=0; counter< size; counter++){
   int element = array[counter];
   total += element;
}
```

=

```
for (int element:array){
     total += element;
}
```

Example:

Write a program that reads a sequence of characters. You want to count the lowercase and the uppercase letters in this sequence. The sequence is finished when % is entered. You should ignore all other characters.

PASSING ARRAYS TO FUNCTIONS

Like any other primitive data types, you can pass arrays to a function.

```
int function_ex(int an_integer, int integer_array[], double double_array[])
```

Example:

Write a program that takes two int values rankA and rankB, then reads the coefficients of two polynomials from the user. The first polynomial has rankA+1 many coefficients, and the second polynomial has rankB+1 many coefficients.

Define a function (addPolynom) that performs the addition operation on these polynomials.

NOTE: Unlike other data types, a function **CANNOT** return an array.

ARRAYS

Example

Example:

Write a program that takes two int values rankA and rankB, then reads the coefficients of two polynomials from the user. The first polynomial has rankA+1 many coefficients, and the second polynomial has rankB+1 many coefficients.

Define a function (multPolynom) that performs the multiplication operation on these polynomials.

SIZE OF AN ARRAY

How to increase the size of an array?

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]	x[10]	x[11]
6	8	45	233	-4	0	4	67	31	3456	34	-69

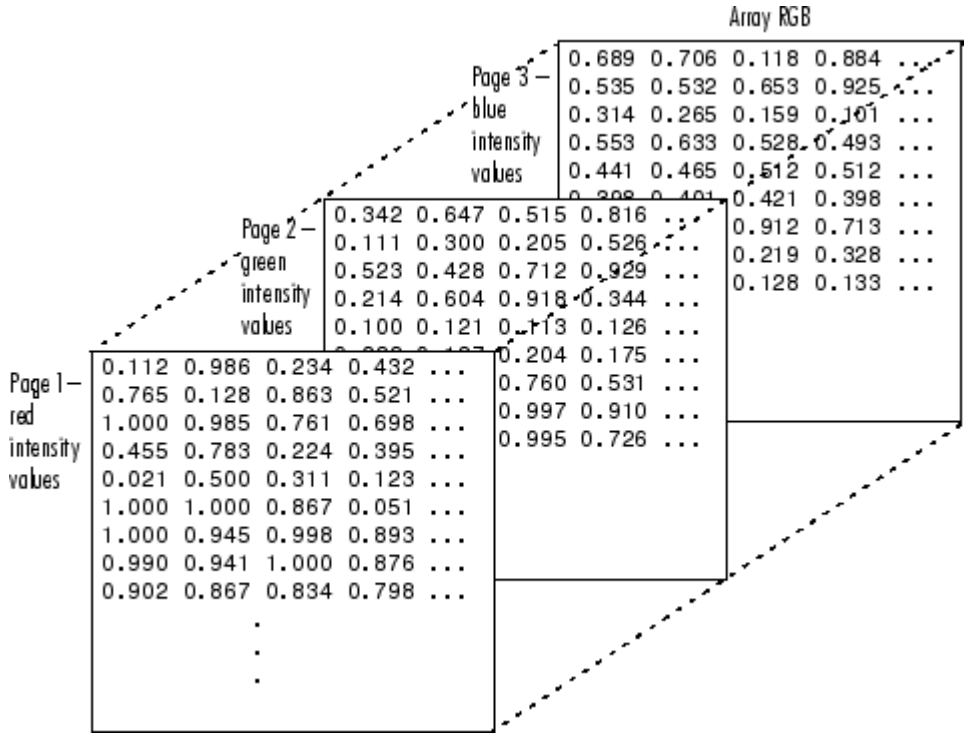
x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]	x[10]	x[11]	x[12]	x[13]
6	8	45	233	-4	0	4	67	31	3456	34	-69	!?	?!

NOTE: Sizes of arrays CANNOT be changed during the program.

"VECTORS"

COMING SOON...

Next week on CMP 1001



MULTIDIMENSIONAL ARRAYS