General Computer Science for Engineers CISC 106 Lecture 34

Dr. John Cavazos
Computer and Information Sciences
05/13/2009

Lecture Overview

- C++ arrays

Declaring of C++ arrays

- A typical declaration for an array is:

type name [elements];

For example:

int grades[60]; // int array of 60 elements

Initializing C++ arrays

- When we declare array we can assign initial values to each element
- Enclose the values in braces { }

For example:

```
int quantity [5] = {10, 235, 77, 40, 12071};
```

Initializing C++ arrays

- Can leave the square brackets empty [ ]
- Array size matches number of elements between {}

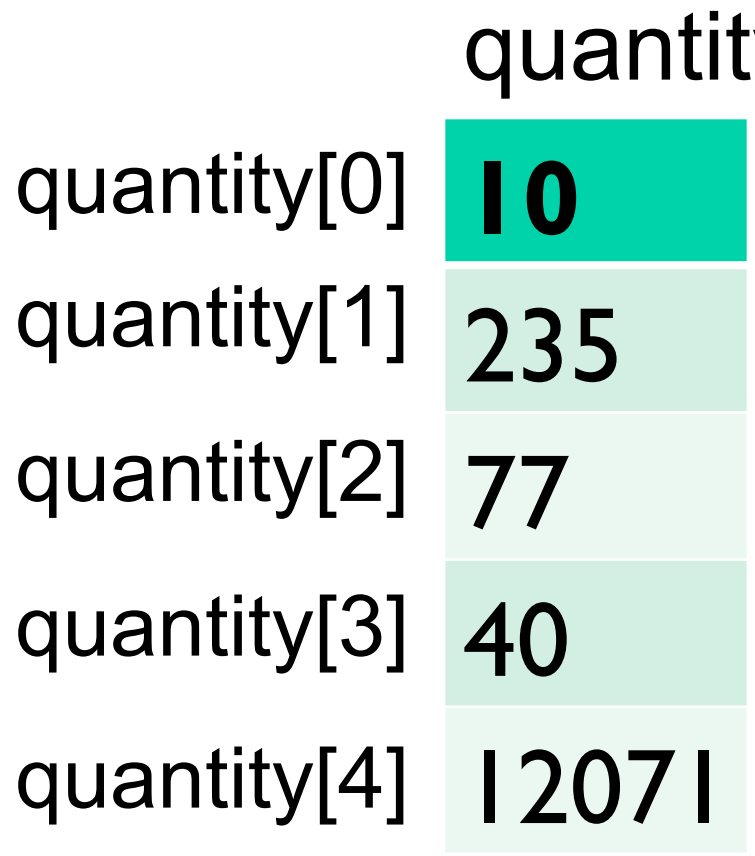
For example:

```
int quantity[] = {10, 235, 77, 40, 12071};
```

C++ arrays have index 0

	quantity
quantity[0]	**10**
quantity[1]	235
quantity[2]	77
quantity[3]	40
quantity[4]	12071

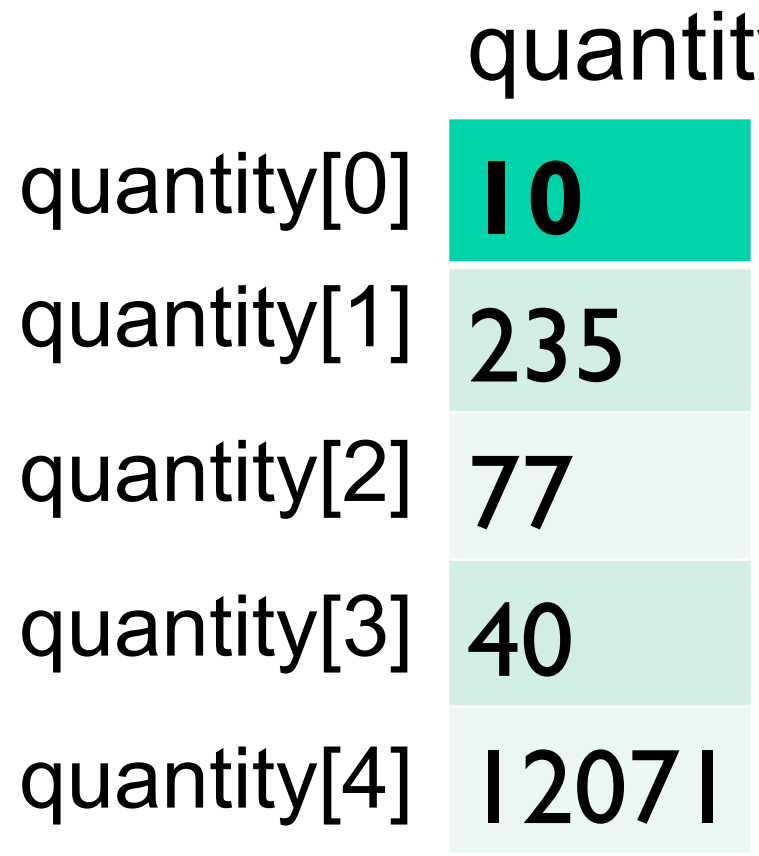
C++ arrays have index 0



A=quantity [0];

After this executes A has value 10.

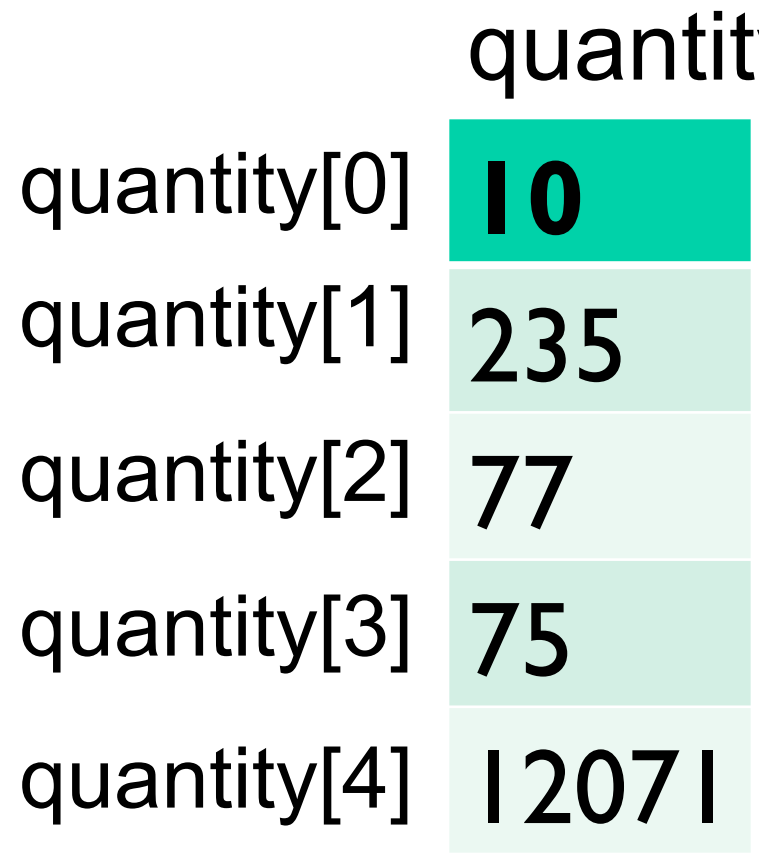
C++ arrays have index 0



quantity [3] = 75;

After this executes quantity[3] will have value 75.

C++ arrays have index 0



Statement has executed.

quantity [3] = 75;

Valid operations with arrays

- quantity[0] = A;
- quantity[A] = 75;
- B = quantity[A+2];
- quantity[quantity[A]] = quantity[2] + 5;

Simple Array Example

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

int main()
{
    int quantity[] = {16, 2, 77, 40, 12071};
    int result = 0;
    for (int i = 0; i<5; i++) {
       result = result + quantity[i];
    }
   cout << result;
}
```

Multidimensional Arrays

- type name [elements]…[elements];

An example of a 5-dimensional array.

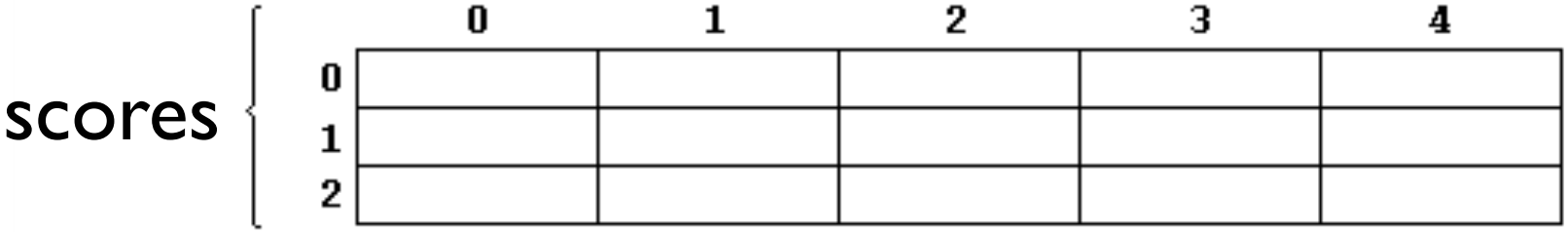
char century[100][365][24][60][60]

Multidimensional Arrays

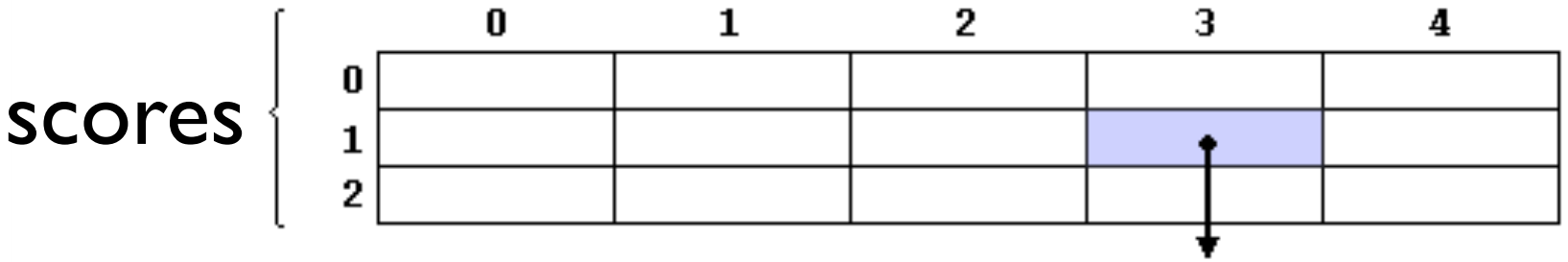
- type name [elements]…[elements];

An example of a 2-dimensional array.

int scores[3][5];



Multidimensional Arrays



scores[1][3]

Arrays as parameters

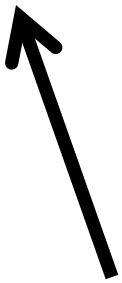
- When declaring a function must specify:
  - The element type
  - An array variable name
  - A pair of (empty) brackets

```
void procedure (int anArray[]) {
    …
}
```

Arrays as parameters

- When declaring a function must specify:
  - The element type
  - An array variable name
  - A pair of (empty) brackets

```
void procedure (int anArray[]) {
   …
}
```

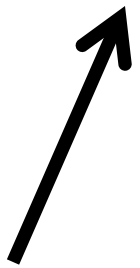


element type

Arrays as parameters

- When declaring a function must specify:
  - The element type
  - An array variable name
  - A pair of (empty) brackets

```
void procedure (int anArray[]) {
   …
}
```

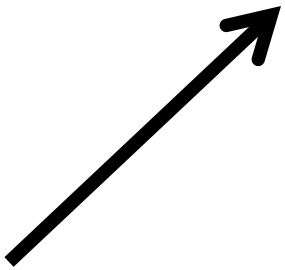


Array variable name

Arrays as parameters

- When declaring a function must specify:
  - The element type
  - An array variable name
  - A pair of (empty) brackets

```
void procedure (int anArray[]) {
    …
}
```



Pair of empty brackets

Example of Passing an Array

```
#include <iostream>
using namespace std;
int sumElements(int [], int);

int main()
{
    int quantity[] = {16, 2, 77, 40, 12071};
    int length = 5;
    int result = sumElements(quantity, length);
    cout << "Sum of Elements: " << result << endl;
}
// rest of program on next slide
```

Example of Passing an Array

```
int sumElements (int anArray[], int length) {
    int sum = 0;
    for (int j =0; j <length; j++) {
       sum = sum + anArray[j];
    }
    return sum;
}
```