FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Review of Introduction to C++

CSPROG2

Computer Programming 2 for CS

Array

An Array is a structured collection of components, all of same type, that is given a single name.

Each component (array element) is accessed by an index that indicates the component's position within the collection.

Array – Defining an Array

Like other variables in C++, an array must be defined before it can be used to store information.

Like other definitions, an array definition specifies a variable type and a name. But it includes another feature i.e. size.

DataType ArrayName [Const Int Expression];

Array Elements

- The items in an array are called array elements.
- All elements in an array are of the same type; only the values vary.

Example

```
int   array1[4] = { 10, 5, 678, -400 } ;
```

Accessing Array Elements

- To Access an individual array component, we write the array name, followed by an expression enclosed in square brackets. The expression specifies which component to access.
- Syntax

  ArrayName [IndexExpression]

- Example

  array1[2]

  array1[i] where i = 2

Initializing Array in Declarations

- To initialize an array, you have to specify a list of initial values for the array elements, separate them with commas and enclose the list within braces.

```
int  array1[5] = {23, 10, 16, 37, 12};
```

- We don't need to use the array size when we initialize all the array elements, since the compiler can figure it out by counting the initializing variables.

```
int array1[ ] = { 23, 10, 16, 37,12};
```

Initializing Array in Declarations

- What happens if you do use an explicit array size, but it doesn't agree with the number of components ?
  - If there are too few components/ items , the missing element will be set to zero.

Lack of Aggregate Array Operations

- C++ does not allow aggregate operations on arrays.

```
int x[50], y[50] ;
```

- There is no aggregate assignment of y to x

```
x = y;   //not valid
```

- To copy array y into array x, you must do it yourself, element by element.

```
for ( i=0; i<50; i++)
        x[i] = y[i];          //valid operation
```

- Similarly, there is no aggregate comparison of arrays.

```
if (x == y )   //Not valid
```

Lack of Aggregate Array Operations

- Also, you cannot perform aggregate input / output operations on arrays.

  cin>>x; //not valid, where x is an array

  cout<<x //not valid

- You cannot perform aggregate arithmetic operations on arrays

  x = x + y // not valid, where x and y are arrays

- Finally, it is not possible to return an entire array as the value of a value-returning function

  return x; //not valid, where x is an array.

Example of One Dimensional Array

```
void main()
  {
        double sales[6], average, total=0;
        cout<< “Enter sales of 6 days”;
        for( int j=0; j<6; j++)
                cin >> sales[ j ];
    for (int j=0; j<6; j++)
                total += sales[ j ] ;
        average = total / 6;
        cout<< “Average =”<< average;
  }
```

Multidimensional Arrays

- A two dimensional array is used to represent items in a table with rows and columns, provided each item in the table is of same data type.
- Each component is accessed by a pair of indexes that represent the component's position in each dimension.

Defining Multidimensional Array

- **Two Dimensional Array**
- The array is defined with two size specifiers, each enclosed in brackets

  DataType ArrayName[ConstIntExp][ConstIntExp]
- Example

  double array2[3][4];
- **Three Dimensional Array**

  float array3[x][y][z]

Accessing Multidimensional Array Elements

- Array elements in two dimensional arrays required two indexes

  array2[1][2]

- Notice that each index has its own set of brackets. Don't write commas.

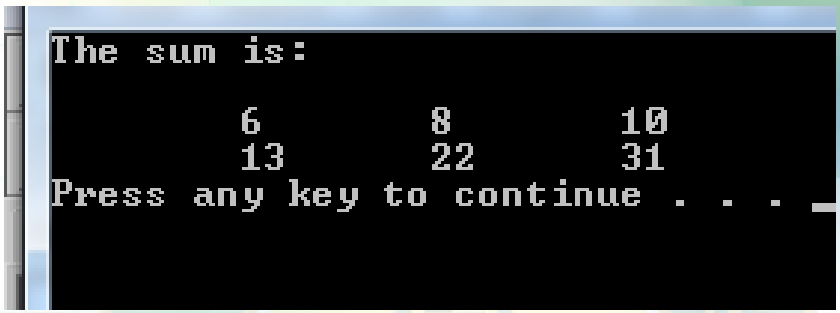
  array2[1,2] // not valid syntax

Sample

```
#include <iostream>
using namespace std;
int main( void ) {
   /* Program to add two multidimensional arrays */
   int a[2][3] = {{5,6,7},{10,20,30}};
   int b[2][3] = {{1,2,3},{3,2,1}};
   int sum[2][3], row, column;

   /* First the addition */
   for(row = 0; row < 2; row++)
      for(column = 0; column < 3; column++)
      sum[row][column]= a[row][column] + b[row][column];
```

Sample




```
/* Then print the results */

cout <<  "The sum is: \n\n" ;
for( row = 0; row < 2; row++ ) {
   for( column = 0; column < 3; column++ )
      cout << "\t" <<  sum[ row ][ column ];
      cout << endl;   /* at end of each row */
   }
system ("pause");
return 0;
}
```