

# Basic Computer Programming and Networking Arrays

*Array:*

23	4	6	15	5	7
0	1	2	3	4	5

↑  
*Array index*

---

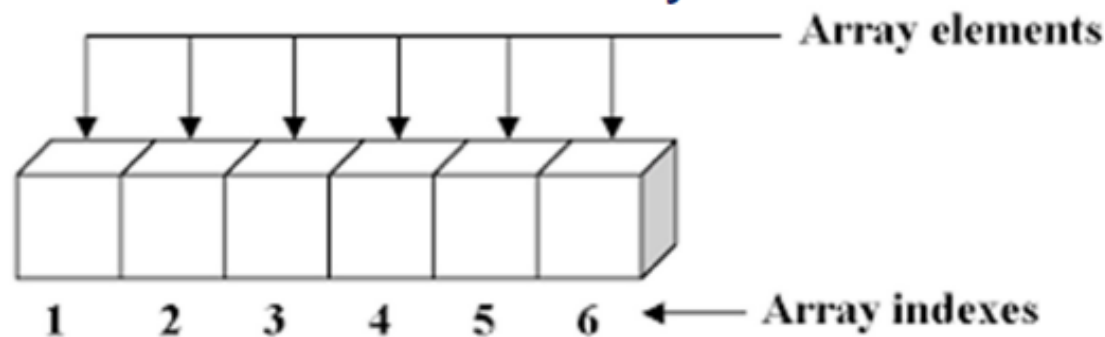
RHNS Jayathissa

Department of Computer Engineering

Faculty Of Computing

# What is Array

- An **array**, is a data structure consisting of a collection of *elements*
- Each identified by at least one *array index* or *key*
- simplest type of data structure is a linear array, also called one-dimensional array



One-dimensional array with six elements

# Arrays Hold Multiple values

- Unlike regular variables, arrays can hold multiple values.

`int count`

Enough memory for 1 `int`

12345

`float price`

Enough memory for 1 `float`

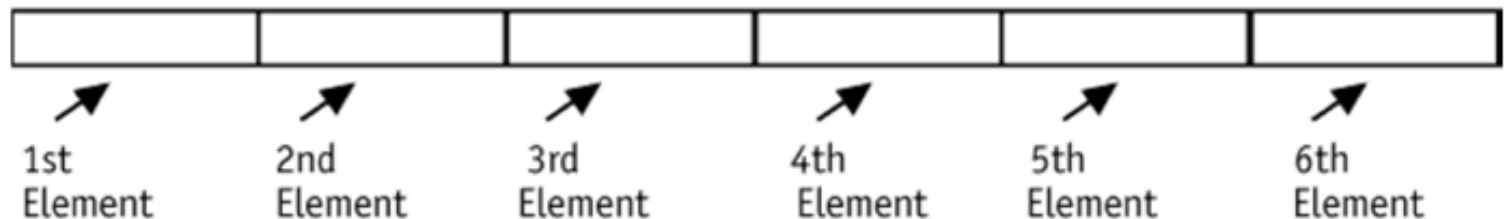
56.981

`char letter`

Enough memory for 1 `char`

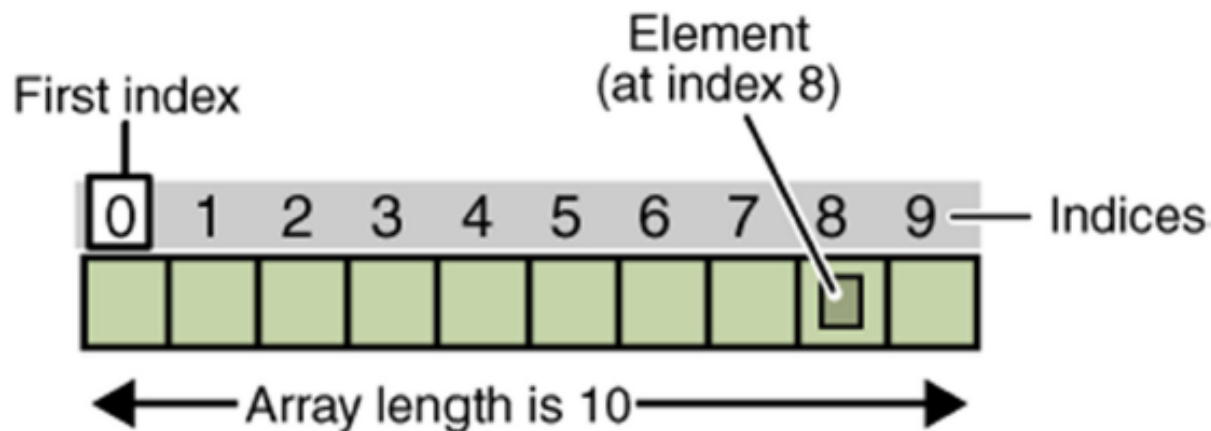
A

Days: enough memory for 6 `ints`



# Array Declaration

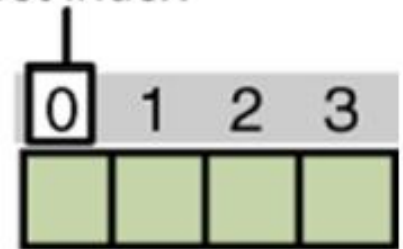
Array Declaration	Number of Elements	Size of Each Element	Size of the Array
<code>char letters[25];</code>	25	1 byte	25 bytes
<code>short rings[100];</code>	100	2 bytes	200 bytes
<code>int miles[84];</code>	84	4 bytes	336 bytes
<code>float temp[12];</code>	12	4 bytes	48 bytes
<code>doubledDistance[1000];</code>	1000	8 bytes	8000 bytes



# Example 1

```
int numbers [4];  
cout << "Enter 4 numbers";  
cin >> numbers[0];  
cin >> numbers[1];  
cin >> numbers[2];  
cin >> numbers[3];
```

First index



```
cout << "Results"<< endl;  
cout << "Number 1 " << numbers[0] << endl;  
cout << "Number 2 " << numbers[1] << endl;  
cout << "Number 3 " << numbers[2] << endl;  
cout << "Number 4 " << numbers[3] << endl;
```

## Example 2

- Enter 50 marks and find the average

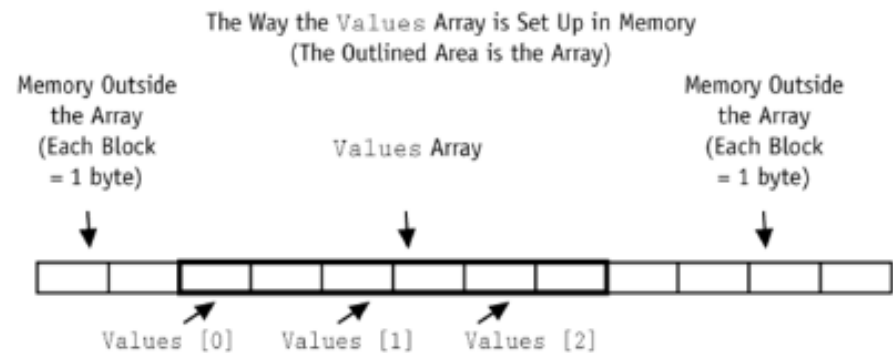
```
int numbers [50];

for(int i = 0; i<50; i++ )
{
    cout << "Enter number ";
    cin >> numbers[i];
}

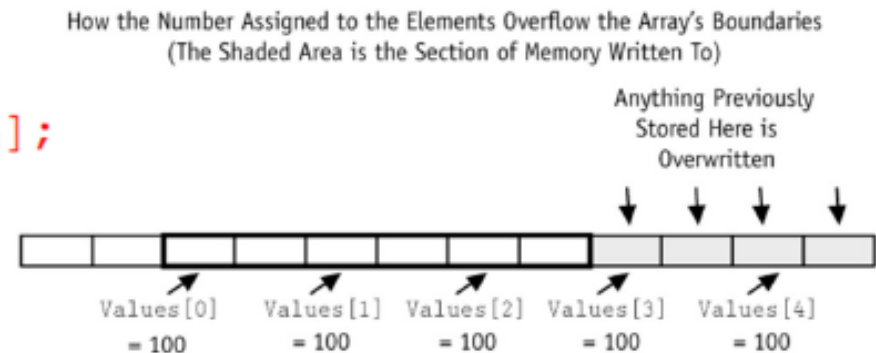
for(int i = 0; i<50; i++ )
{
    cout << "Number "<< numbers[i];
}
```

# No Bounds Checking in C++

- C++ gives you the freedom to store data past an array's boundaries.



```
int numbers [3];  
cin >> numbers[4];  
cout << "value " << numbers[4];
```



# Array Initialization

- Arrays may be initialized when they are declared.

```
int numbers [3];  
numbers[0] = 10;  
number [1] = 5;
```

- When an array is being initialized, C++ does not require a value for every element.

```
int numbers[7] = {1, 2, 4, 8};
```

- It is possible to declare an array without specifying its size, as long as you provide an initialization list.

```
int numbers[] = {1, 2, 4, 8};
```



# Initializing With Strings

- When initializing a character array with a string, simply enclose the string in quotation marks:

```
char Name [7] = "Warren";
```

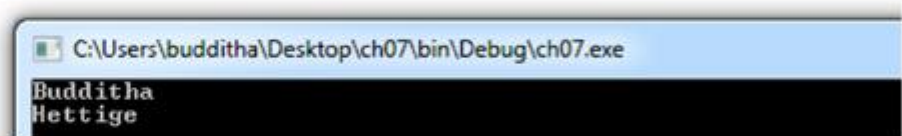
Null Terminator



'W'	'a'	'r'	'r'	'e'	'n'	'\0'
Name [0]	Name [1]	Name [2]	Name [3]	Name [4]	Name [5]	Name [6]

# Example 3

```
char name1[] = "Budditha";  
char name2[] = {'H', 'e', 't', 't', 'i', 'g', 'e', '\\0'};  
  
cout << name1 << endl;  
cout << name2 << endl;
```



# More on Arrays

- Individual array elements are processed like any other type of variable.
- By using the same subscript, you can build relationships between data stored in two or more arrays.
- You cannot use the assignment operator to copy one array's contents to another.

# Printing the Contents of an Array

- To display the contents of an array, you must use a loop to display the contents of each element.

```
int array[5] = { 10, 20, 30, 40, 50 };  
  
for (int count = 0; count < 5; count++)  
{  
    cout << array[count] << endl;  
}
```

# Exercise

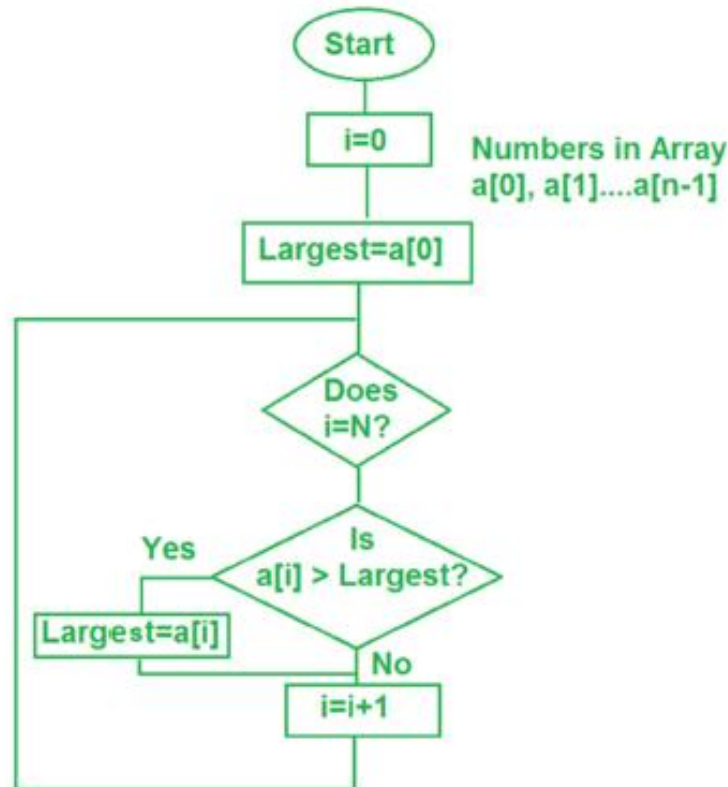
- Write a C++ program that read 10 integer marks (Range 0 to 100) form keyboard and print the result sheet as shown below.

```
ICT 126 2.0 Introduction to Computer Programming
-----
#####          RESULT SHEET          #####
-----
Index NO          Marks
=====
1                  56
2                  79
3                  45
...

```

# Find elements

- Find the largest and small mark of the above marks array



# Example (several arrays)

- Create 2 Arrays to store Student name and GPA values of the 25 students in a class
- Enter all values and print results in the following format

```
Student summary
```

```
-----
```

```
Saman Kumara      2.56
```

```
Ruwan Silva       1.94
```

# Multiple Arrays

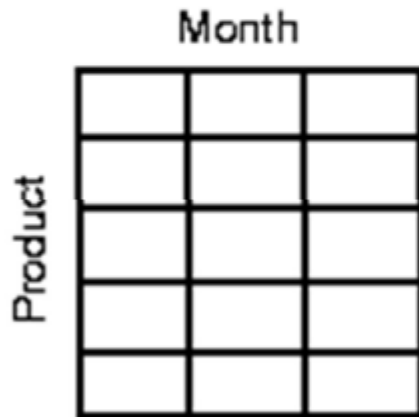
- Write a C++ program to store 10 student information in class with considering name, age, height. Your program should read 10 students information from keyboard and display results as given below

name	age	height
-----		
saman	14	5.7
kamal	12	5.1



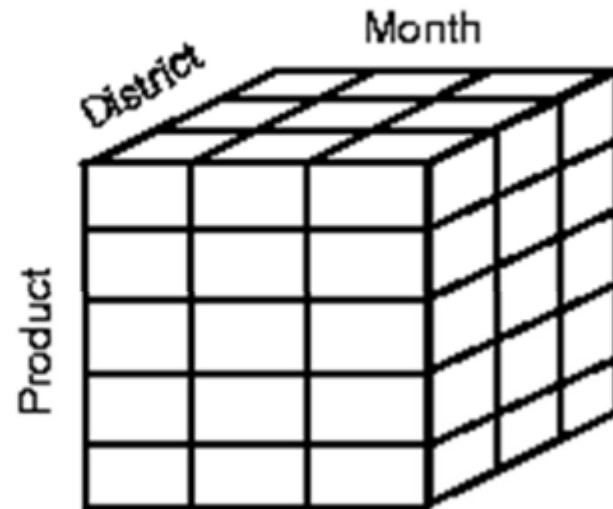
# Multi-dimensional arrays

Two-Dimensional  
Spreadsheet



Sales data for each district is in a separate spreadsheet

Multidimensional Array



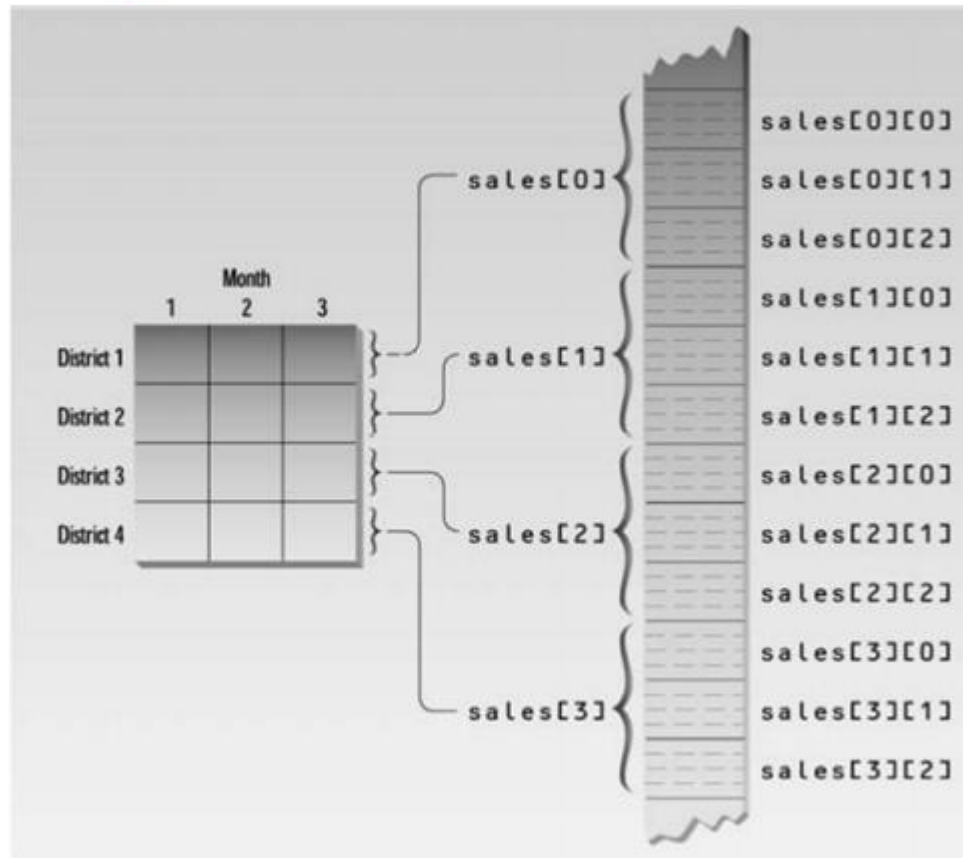
Sales data for all districts is in a single array

# Two-dimensional Arrays

- A two-dimensional array is like several identical arrays put together. It is useful for storing multiple sets of data.

```
float sales[3][4];  
// 2D array, 3 rows and 4 columns.
```

# 2D Array



# Example

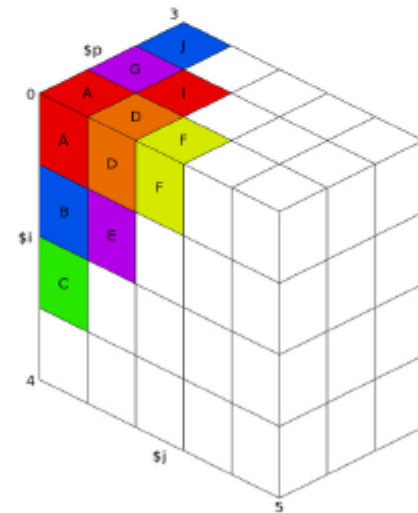
- Write a C++ program to display the following Multiplication table

1	2	3	4	5	6	7	8	9
-----								
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	13	16	20	24	28	32	36
.								
.								
.								
9								81

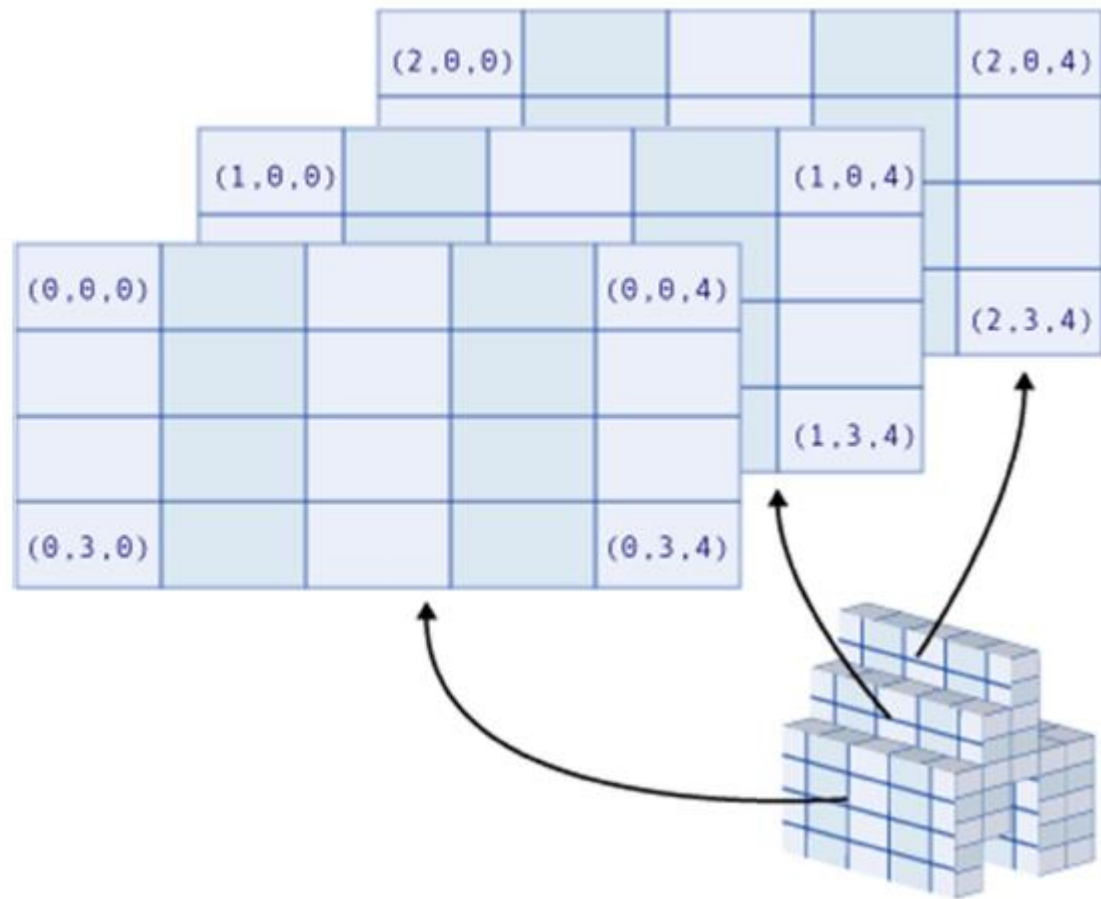
# Three Dimensional Arrays and Beyond

- C++ allows you to create arrays with virtually any number of dimensions.
- Here is an example of a three-dimensional array declaration:

```
float seat[3][5][8];
```



# 3D Array



# Exercise

- Write a function called `reversit()` that reverses a C-string (an array of `char`). Use a for loop that swaps the first and last characters, then the second and next-to-last characters, and so on. The string should be passed to `reversit()` as an argument.
- Write a program to exercise `reversit()`. The program should get a string from the user, call `reversit()`, and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase,

“Able was I ere I saw Elba.”

# Exercise

- Start with a program that allows the user to input a number of integers, and then stores them in an int array. Write a function called `maxint()` that goes through the array, element by element, looking for the largest one. The function should take as arguments the address of the array and the number of elements in it, and return the index number of the largest element. The program should call this function and then display the largest element and its index number.



---

# Thank You