VTC Academy
Think Ahead, Beyond Boundaries

BASIC PROGRAMMING LANGUAGE

LESSON 6

Arrays

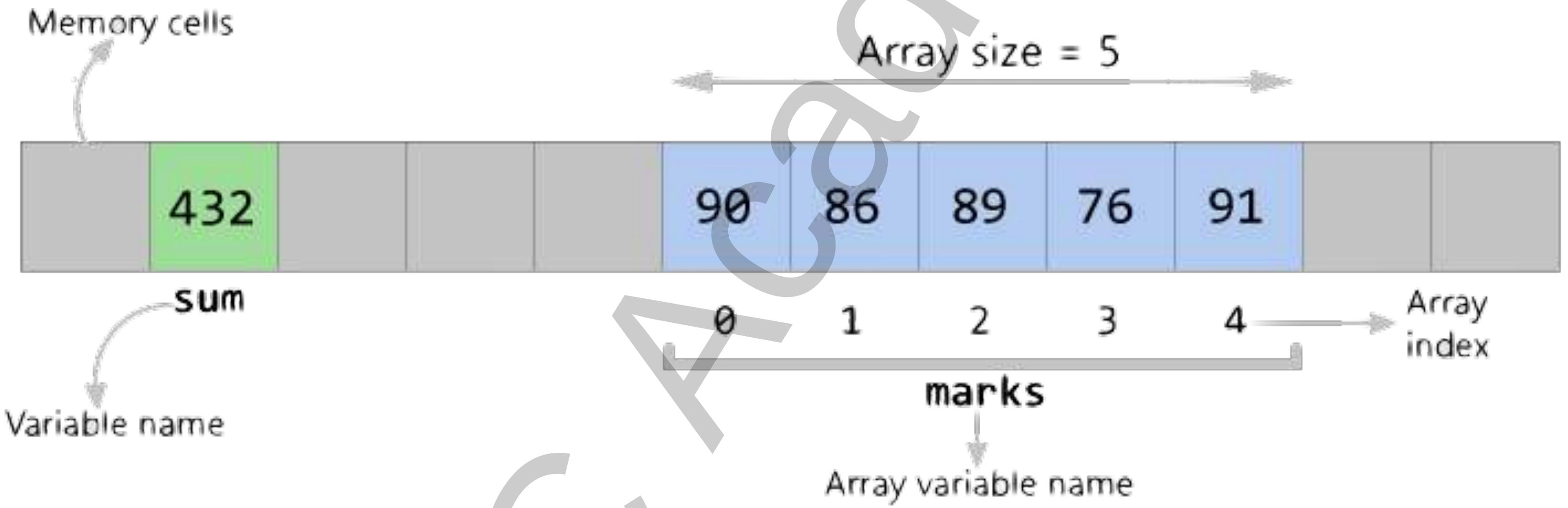
CONTENT

1. Introduction to Array
2. Explain Array Elements and Indices
3. Initialize Array
4. Introduction to Multi-dimensional Array
5. Single Dimensional Array Handling (Insert / Update / Display)
6. Summary

What is Arrays?

- Array is a kind of data structure that can store a fixed-size sequential collection of elements of the same type.
- Each element of the array has the same data type, same storage class and same characteristics.
- These elements are known as members of the array.
- Arrays can be used to store collection of primitive data types such as int, float, double, char, etc of any particular type.

What is Arrays?



Why Do We Need Arrays?

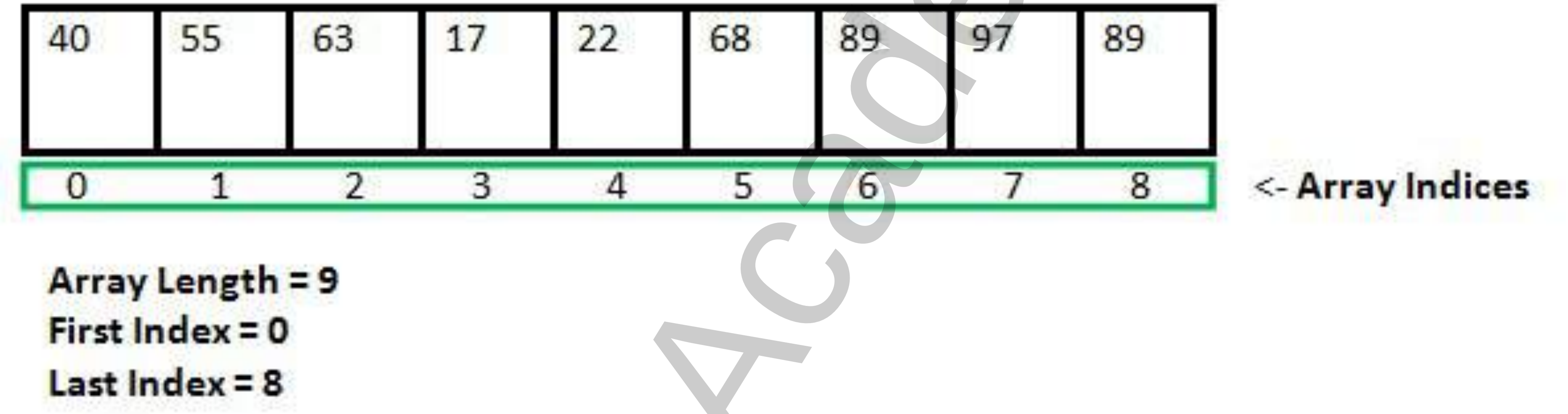
- Normal variables (v1, v2, v3,…) are useful when we have a small number of data items.
- However, if we want to store a large number of items, it becomes difficult to manage them with normal variables. The idea of an array is to represent many instances in one variable.
- Arrays help us easy to handle a collection of the same type data items.

Array Elements & Indexes

- Each member of an array is identified by unique index or subscript assigned to it.
- An index is a positive integer enclosed in [ ] placed immediately after the array name.
- An index holds integer values starting with zero (0).
- Example of an array named “players” with 9 elements will look like:

```
players[0], players[1], …, players[8]
```

Array Elements & Indexes



40	55	63	17	22	68	89	97	89	
0	1	2	3	4	5	6	7	8	<- Array Indices

Array Length = 9

First Index = 0

Last Index = 8

Array Declaration

- An array has some particular characteristics and has to be defined with them.
- These characteristics include:
  - Data Types - value types of the elements in the array
  - Array Name - indicates the location of the first member of the array
  - Array Size - a constant evaluating to a value

Array Declaration

- To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows:

```
type array_name [array_size];
```

- This is called a single-dimensional array.
- The `array_size` must be an integer constant greater than zero and type can be any valid C data type.
- Example:

```
int players[9];
```

Initializing Arrays

- You can initialize an array in C either one by one or using a single statement as follows:

```
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

- The number of values between braces { } cannot be larger than the number of elements that we declare for the array between square brackets [ ].
- If you omit the size of the array, an array just big enough to hold the initialization is created:

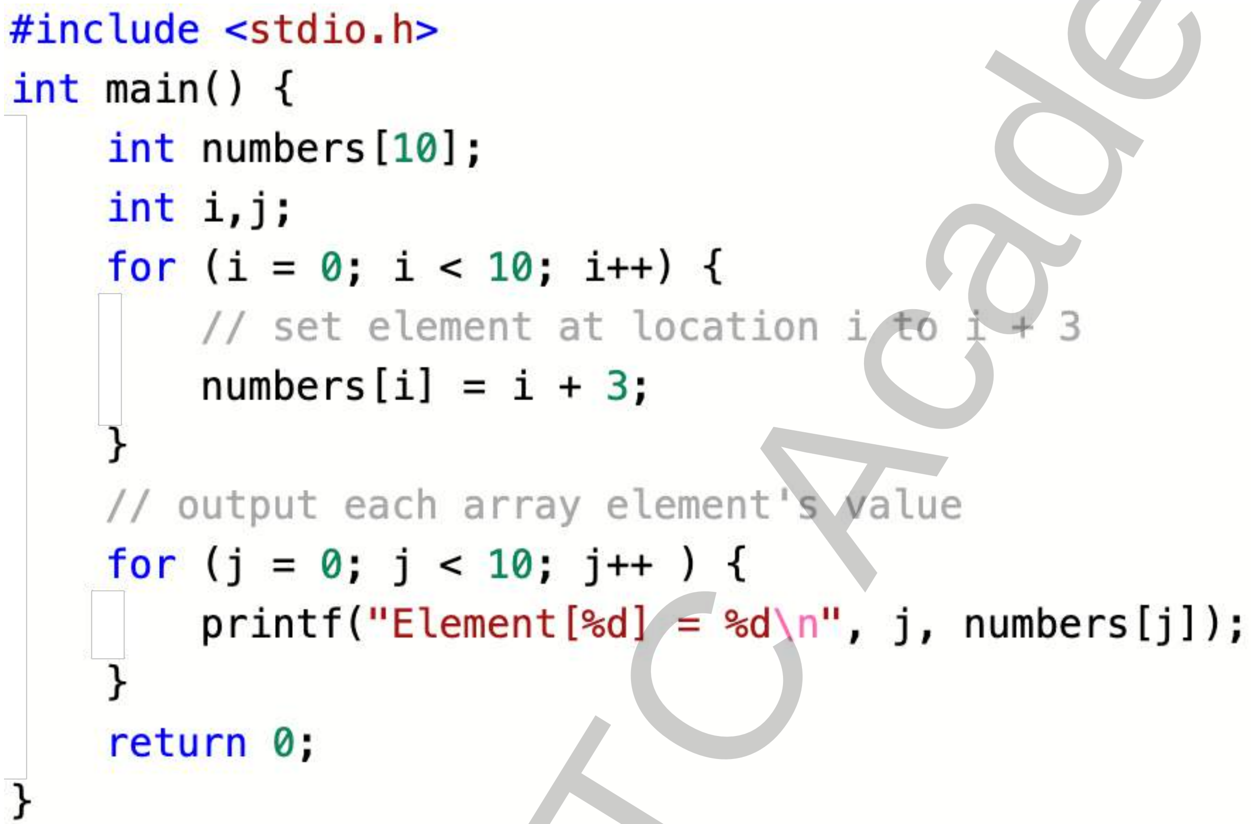
```
double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

Accessing Array Elements

- An element is accessed by indexing the array name.
- This is done by placing the index of the element within square brackets after the name of the array.
- You can assign the element of array to a particular value and get the value of each element in the array as follow:

```
balance[4] = 50.0;  // assign 4th element to 50.0
double salary = balance[9];  // get the value of 9th element
```

Accessing Array Example



```c
#include <stdio.h>
int main() {
    int numbers[10];
    int i,j;
    for (i = 0; i < 10; i++) {
        // set element at location i to i + 3
        numbers[i] = i + 3;
    }
    // output each array element's value
    for (j = 0; j < 10; j++ ) {
        printf("Element[%d] = %d\n", j, numbers[j]);
    }
    return 0;
}
```

Multi-dimensional Array

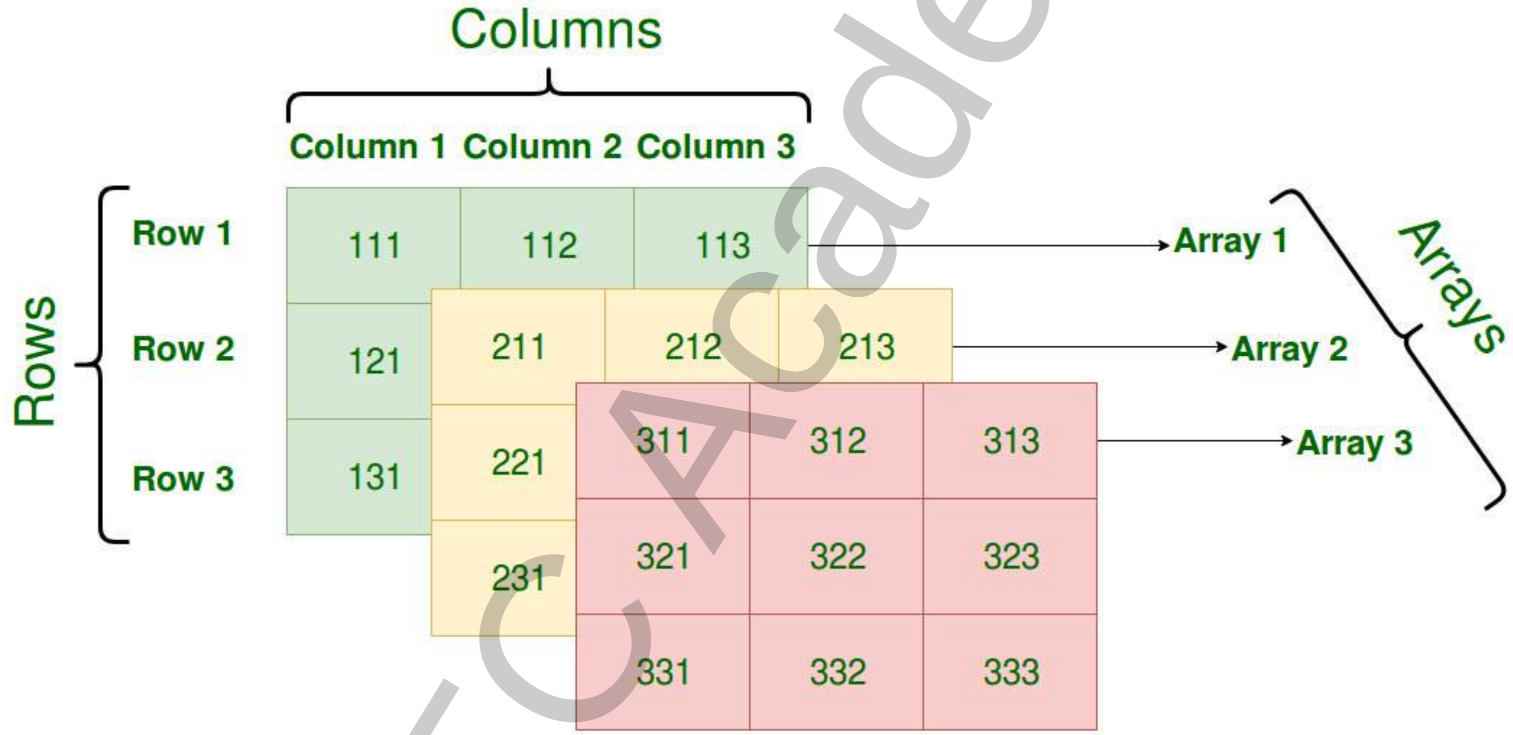
- C programming language allows multidimensional arrays.
- Syntax:

```
type name[size1][size2]...[sizeN];
```

- Example:

```
int 3darray[5][10][4];
```

Multi-dimensional Array

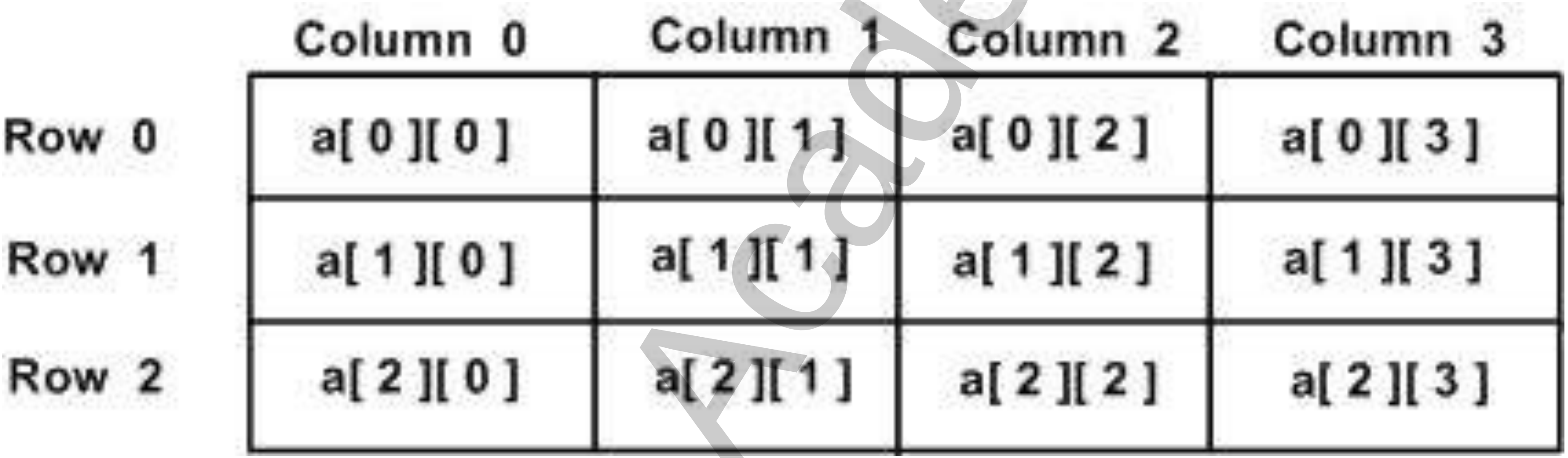


Multi-dimensional Array

- The simplest and the most commonly used multi-dimensional array is the two-dimensional array.
- A two-dimensional array can be thought of as an array of two single dimensional arrays.
- A two-dimensional array looks like a railway time-table consisting of rows and columns.
- A two-dimensional array is declared as

```
int numbers[4][3];
int numbers[3][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
int numbers[3][4] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 3}};
```

Multi-dimensional Array



	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Accessing Two-dimensional Array

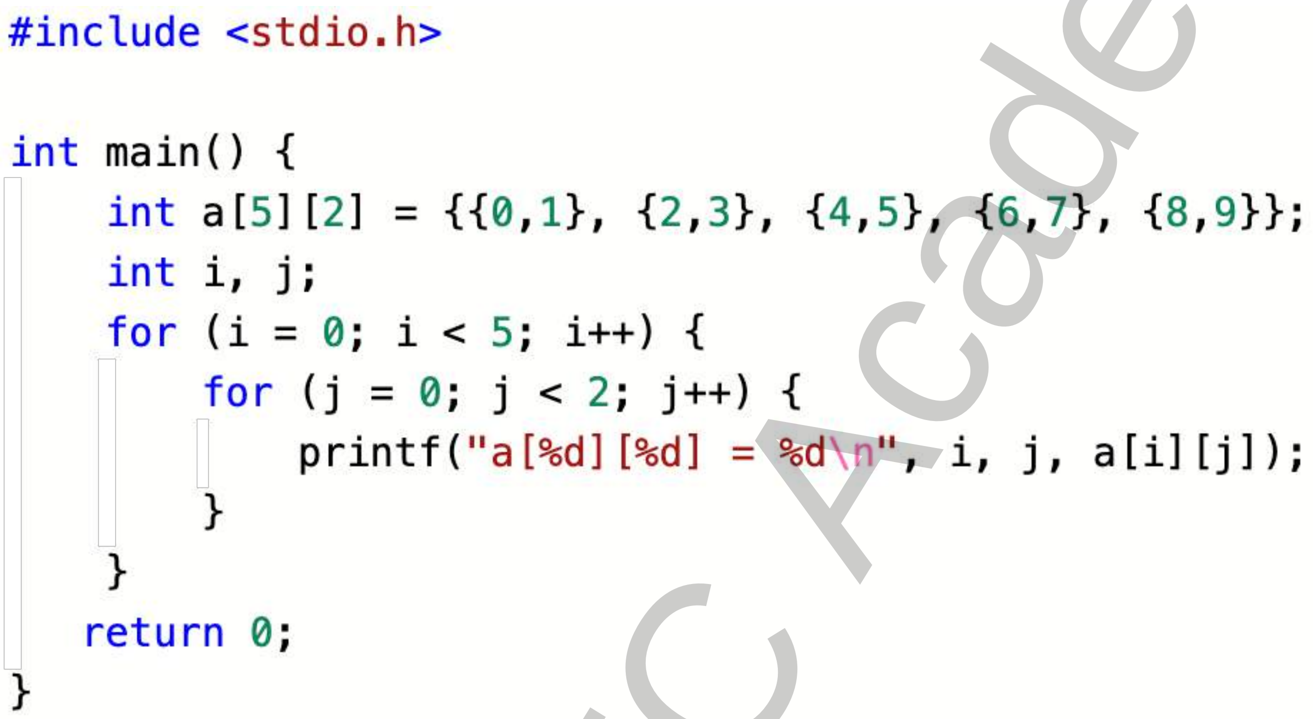
- Initializing two-dimensional array:

```
int a[3][4] = {
    {0, 1, 2, 3} , /* initializers for row indexed by 0 */
    {4, 5, 6, 7} , /* initializers for row indexed by 1 */
    {8, 9, 10, 11} /* initializers for row indexed by 2 */
};
```

- An element in a two-dimensional array is accessed by using the subscripts, i.e., row index and column index of the array:

```
int val = a[2][3];
```

Two-dimensional Array Example



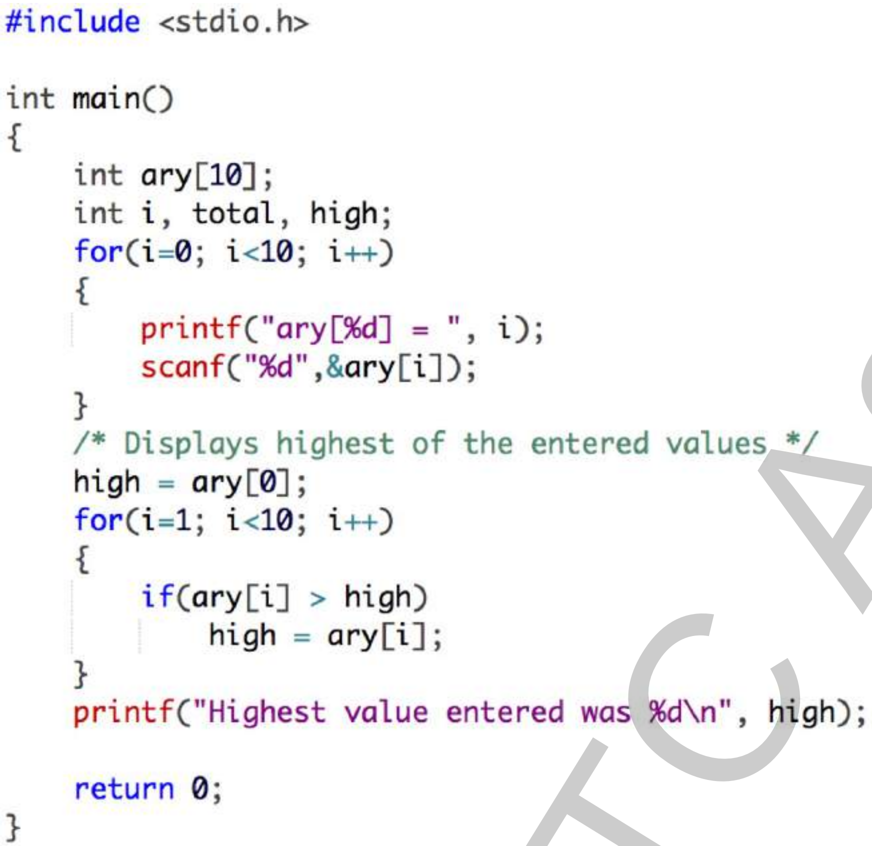
```
#include <stdio.h>

int main() {
    int a[5][2] = {{0,1}, {2,3}, {4,5}, {6,7}, {8,9}};
    int i, j;
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 2; j++) {
            printf("a[%d][%d] = %d\n", i, j, a[i][j]);
        }
    }
    return 0;
}
```

Array Handling

- An array is treated differently from a variable in C.
- Two arrays, even if they are of the same type and size cannot be tested for equality.
- It is not possible to assign one array directly to another.
- Values cannot be assigned to an array on the whole, instead values are assigned to the elements of the array.
- Arrays often are used to store a collection of primitive value types and programmers can easily to implement sort and search algorithm to manipulate with arrays.

Array Handling Example



```c
#include <stdio.h>

int main()
{
    int ary[10];
    int i, total, high;
    for(i=0; i<10; i++)
    {
        printf("ary[%d] = ", i);
        scanf("%d",&ary[i]);
    }
    /* Displays highest of the entered values */
    high = ary[0];
    for(i=1; i<10; i++)
    {
        if(ary[i] > high)
            high = ary[i];
    }
    printf("Highest value entered was %d\n", high);

    return 0;
}
```

Array Handling Demo

Insert / Update / Display with 2D Array

Summary

- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.
- Each element of the array has the same data type, same storage class and same characteristics.
- An index is a positive integer enclosed in [ ] placed immediately after the array name. An index holds integer values starting with zero.
- An element is accessed by indexing the array name.
- This is done by placing the index of the element within square brackets after the name of the array.

VTC Academy
Think Ahead, Beyond Boundaries

Thank you!