

**Eastern Mediterranean University**  
**School of Computing and Technology**  
**Information Technology**  
**Lecture3 – Arrays**

**One Dimensional Arrays**

**Why do we need arrays?**

- There are 800 salaried employees in a company. You need to read the 800 salaries from the input, compute the average salary, and determine how many employees are paid above average.
- You cannot possibly compute all this while reading the salaries from the input. The 800 salaries must be stored somewhere.

Maybe you can do this?

```
double salary1, salary2, ..., salary800;
```

**X** But these variables cannot be processed in a loop

**Solution: Arrays**

**What's an array?**

An array is ordered lists of variables of the same type that can be referenced by subscripting a single identifier with a variable index,

for example like this: **salaries[i]**

- If an array y has n elements, then:
  - the "first" element is: y[0]
  - the last is: y[n-1].

Example:

```
int y[4];
```

contains the following elements:

y[0], y[1], y[2] and y[3]

Note: y[4] is NOT part of the array!!

**Declaring an array:**

Syntax: type identifier[number\_of\_elements];  
**double salaries[800];**

**Referencing an array:** an element of an array is accessed by using the subscript operator. The syntax is: `array_name[index/subscript number]`  
`salaries[0], salaries[i], salaries[2*i+j], salaries[exp]`

### Notes on Arrays in C:

- The array elements are always stored in consecutive memory cells.
- The index of the first element in an array is always 0.
- The number of elements in an array must be a constant. For example, you cannot do this:

```
int array_size;  
scanf("%d",&array_size);  
int my_array[array_size];
```

Example:

```
int x = 200;  
int y[x]; → WRONG!!!!!!
```

### Initialization of Arrays:

- Arrays can be initialized in a loop, like this:

```
for (i = 0; i < 800; i++)  
    scanf("%lf", &salaries[i]);
```

```
for (i = 0; i < 800; i++)
```

- Arrays can be also initialized at declaration time, like this:

```
double prices[3] = {7.49, 9.99, 1.49};
```

equivalent



```
int grades[5] = {100, 97, 79, 0, 0};
```

```
int grades[5] = {100, 97, 79};
```

- Compiler can determine array size from declaration-time initialization:

equivalent



```
int grades[5] = {100, 97, 79, 0, 0};
```

```
int grades[] = {100, 97, 79, 0, 0};
```

### example:

```
int A[SIZE] = {7, -9}; // A[2] and A[3] are set to 0  
int counts[100] = {0}; // initialize 100 integers to 0
```

```
double average[12] = {0.0}; // initialize 12 doubles to 0.0
```

## Examples:

```
#define N 50

int A[N]; // assume initialization
int i;

// assign 7 to the first element and -25 to the last element in array A
A[0] = 7;
A[N-1] = -25; // better than A[49] = -25;

//read in array elements with odd indexes
for( i = 1; i < N; i += 2)
    scanf("%d",&A[i]);

//print out the 4th through 9th elements, position 3 through position 8
for( i = 3; i < 9; i++)
    printf("%d",A[i]);

//print out all positive values one per line:
for(i = 0; i < N; i++)
{
    if(A[i] > 0)
        printf("%d\n",A[i]);
}

//print out the position of all positive values one per line:
for(i = 0; i < N; i++)
{
    if(A[i] > 0)
        printf("%d\n",i);
}
```

## Constant Declaration

```
#define ACONSTANT 300.0
```

```
#define TRUE 1
```

```
#define FALSE 0
```

Note: NO semicolon at the end!

the name of a constant is usually written in uppercase.

**Advice:** always named constants for the sizes of the arrays. It makes it easier to write, read and maintain the code.

## Example: finding average of even numbers

```
#include "stdafx.h"
```

```
#define MAX 300
```

```

void main()
{
    int v, x[ MAX ], num=2;
    float accum = 0, average;

    for(v = 0; v < MAX; v++)
    {
        x[ v ] = num;
        num+=2;
    }

    for(v = 0; v < MAX; v++)
        accum += x[v];

    average = accum / MAX;
    printf("Average=%.2f",average);
}

```

### Example: sort numbers in ascending order

```

#include "stdafx.h"
#define MAX 15

void main()
{
    int v,x, volt[ MAX ],temp;

    for(v = 0; v < MAX; v++)
    {
        scanf("%d",&volt[v]);
    }

    for( int y = 0; y < MAX - 1; y++)
    {
        for(int x = 0; x < MAX - 1 - y; x++)
        {
            if( volt[x] > volt[x + 1] )
            {
                temp = volt[x];
                volt[x] = volt[x + 1];
                volt[x + 1] = temp;
            }
        }
    }

    for(v = 0; v < MAX; v++)
        printf("%3d\n",volt[v]);
}

```

### Example: Working with Salaries

```

#include "stdafx.h"
#define NUM_EMPLOYEES 5
void main()
{
    double salaries[NUM_EMPLOYEES], sum = 0.0, average;

```

```

int i, number_above_average = 0;
for (i = 0; i < NUM_EMPLOYEES; i++)
{
    scanf("%lf", &salaries[i]);
    sum += salaries[i];
}
average = sum/NUM_EMPLOYEES;

for (i = 0; i < NUM_EMPLOYEES; ++i)
{
    if(salaries[i]>average)
        number_above_average++;
}
// number_above_average += (salaries[i] > average);

printf("The average is: %.2f\n", average);
printf("There are %d salaries above the average\n", number_above_average);
}

```

### Example: Computing the Histogram

The input consists of  $n = 20$  exam grades, which are integers in the range  $0, 1, \dots, 100$ . How many times does each grade occur?

```

#include "stdafx.h"
#define NUM_OF_STUDENTS 5
void main()
{
    int histogram[101] = {0};
    int i, current_grade;
    for (i = 0; i < NUM_OF_STUDENTS; i++)
    {
        scanf("%d", &current_grade);
        histogram[current_grade]++;
    }
    printf("%5s%12s\n", "GRADE", "FREQUENCY");
    for (i = 1; i <=100; i++)
        printf("%5d%12d\n", i, histogram[i]);
}

```

### Example: printing 5 random numbers from 0 to 49

```

#include "stdafx.h"
#include <stdlib.h> /* srand, rand */
#include <time.h> /* time */

void main ()
{
    int i, n;
    /* Intializes random number generator */
    srand (time(NULL));

    n = 5;

    /* Print 5 random numbers from 0 to 49 */
    for( i = 0 ; i < n ; i++ ) {
        printf("%d\n", rand() % 50);
    }
}

```

**Ex:** Count Occurrences of Specified Value in an Array

```
#include "stdafx.h"
#define SIZE 7
void main()
{
    int i;
    int frequency[6] = { 0 };
    int responses[ SIZE ] = {2,1,5,4,3,3,2};

    for ( i = 0; i < SIZE; i++ )
        ++frequency[ responses [ i ] ];

    printf( "%7s%17s\n", "Numbers", "Frequency" );

    for ( i = 1; i <=5; i++ )
        printf( "%7d%17d\n", i, frequency[ i ] );
}
```

**Ex:** Passing array to Function

```
#include "stdafx.h"
#include "stdlib.h"
#include "time.h"
#define SIZE 10
void funcArray(int[]);
void main()
{
    int i;
    int num[SIZE] ;
    srand(time(NULL));
    for(int i=0;i<SIZE;i++)
        num[i]=rand()%20+5;
    funcArray(num);
}
void funcArray(int num[])
{
    for(int i=0;i<SIZE;i++)
        printf("%4d",num[i]);
}
```

**Ex:** inserting an element in an array

```
#include "stdafx.h"
void main()
{
    int a[10] = {5,2,4,3}, x, loc, i;

    for (i = 0; i < 4; i++)
        printf("%3d",a[i]);

    printf("\n\nenter element:");
    scanf_s("%d", &x);
}
```

```
printf("enter location:");
scanf_s("%d", &loc);

for (i = 4; i >loc; i--)
    a[i] = a[i-1];

a[loc] = 6;
for (i = 0; i <5; i++)
    printf("%3d",a[i]);
}
```