Array

Prepared by: Lec Tasmiah Tamzid Anannya, CS Dept, AIUB

What is Array?

- Array: a set of ordered data items.
- Array is a kind of data structure that can store a fixed-size sequential collection of elements of the same type.
- You can define a variable called x, which represents not a *single* value, but an entire *set of values*.
- Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables.

What is Array?

- Each element of the set can then be referenced by means of a number called an *index* number or *subscript*.
- Mathematics: a subscripted variable, x_i, refers to the *i*th element *x* in a set
- C programming: the equivalent notation is x[i]
- A specific element in an array is accessed by an index.

Number[0]	Number[1]	Number[2]	Number[3]	Number[4]	……..

Declaring Arrays

- 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 arrayName [arraySize]

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

Declaring Arrays

- 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 arrayName [arraySize]

- This is called a *single-dimensional* array.
- The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type.
- For example:

double myarray[10]

char name[20]

What happens when an array is declared?

- double myarray[10]
- We can access these 10 elements individually by:

myarray[0]

myarray[1]

myarray[2]

.

.

.

myarray[9]

Remember, index of array start at 0. So, an index of 1 references the second element of the array.

Initializing array

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

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

Initializing array

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

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

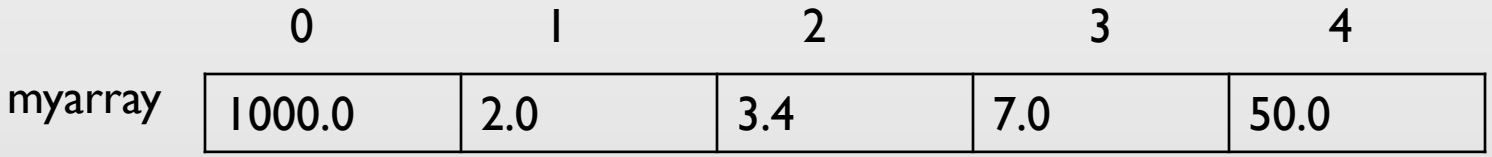
- Or, ***myarray[0]=1000.0***

Initializing array

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

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

- Or, ***myarray[0]=1000.0***



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.
- For example-

double a=myarray[0];

- The above statement will take the 1st element from the array and assign the value to variable named ***a***.

Example

```
#include <stdio.h>
int main()
{
    int arr[4]; /* arr is an array of 4 integers */
    arr[0] = 5;   //initializing 1st element
    arr[2] = -10; //initializing 2nd element
    arr[1] = 2; //initializing 3rd element
    arr[3] = arr[0]; //initializing 4th element

    printf("%d %d %d %d", arr[0], arr[1], arr[2], arr[3]);

    return 0;
}
```

Example

```
#include <stdio.h>
int main () {

   int n[ 10 ]; /* n is an array of 10 integers */
   int i,j;

   for ( i = 0; i < 10; i++ ) {
      n[i] = i + 100; /* set element at location i to i + 100 */
   }
   /* output each array element's value */
   for (j = 0; j < 10; j++ ) {
      printf("Element[%d] = %d\n", j, n[j] );
   }
   return 0;
}
```

Example

```
#include <stdio.h>
int main () {

   int n[ 10 ]; /* n is an array of 10 integers */
   int i,j;

   for ( i = 0; i < 10; i++ ) {
      n[i] = i + 100; /* set element at location i to i + 100 */
   }
   /* output each array element's value */
   for (j = 0; j < 10; j++ ) {
      printf("Element[%d] = %d\n", j, n[j] );
   }
   return 0;
}
```

```
Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109
```

Taking input of an array

scanf("%d", &myarray[0]);

- C does not perform bound checking on array indexing.
- It is possible to overrun the end of an array.
- Suppose, an array a is declared having 4 element,

  int a[4];
- The compiler will still let you access the 10th member by a[9].
- Of course, attempting non existent members will have disastrous results.
- So the programmers must be careful.

Practice

- Now try to declare an array of size 5 and take input for this array and print all the elements of the array USING LOOP.

Practice

- Now try to declare an array of size 5 and take input for this array and print all the elements of the array USING LOOP.

```
#include <stdio.h>
int main()
{
int i;
   int myarray[5];
   for(i=0;i<5;i++) //for taking input
   {
       scanf("%d", &myarray[i]);
   }
   for(i=0;i<5;i++) //for printing the elements of the array
   {
       printf("%d\n", myarray[i]);
   }
    return 0;
}
```

- In C, you can not assign one entire array to another.

```
int myarray[5], a[5];
a=myarray;          //ERROR
```

Practice

- Try to copy an array to another using a loop.

- Declare an array of integer for 5 numbers and calculate the sum of them.