ARRAYS
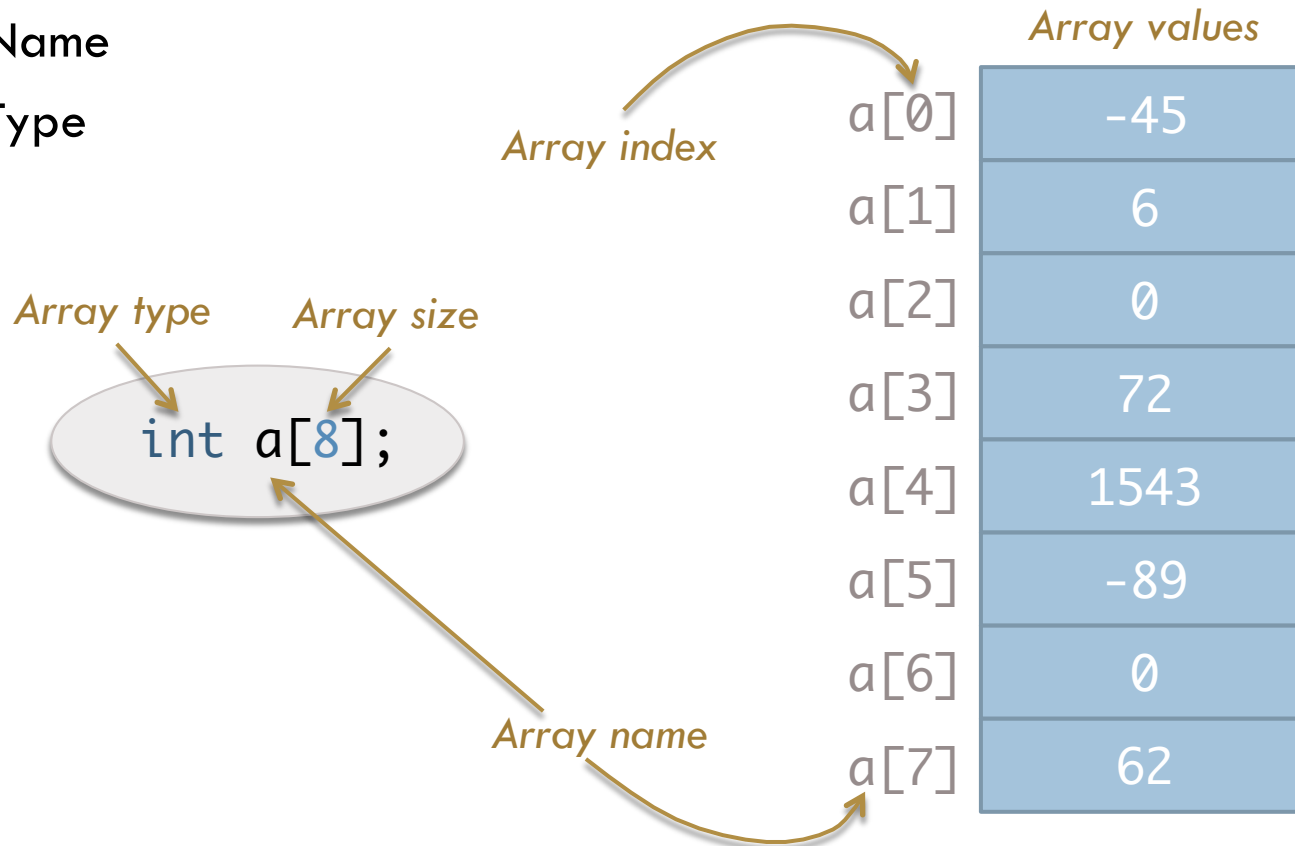
Section 1D

What's an array?

- A group of memory locations that are related to one another; they share
  - Name
  - Type

Array type *Array size*

```
int a[8];
```

Array name

Array index

Array values

a[0]	-45
a[1]	6
a[2]	0
a[3]	72
a[4]	1543
a[5]	-89
a[6]	0
a[7]	62

How do we work with arrays?

- Indices are between 0 and SIZE-1

```
int b = 1, c = 3;
a[b + c] = 1;      // a[4] = 1.
a[8] = 2;          // Error! Index out of bounds.
cout << a[4];      // Prints 1.
```

- Creating an array

```
double d[5];              // 5 undefined double values.
double d[3] = {1,2,3};    // 3 double values: 1, 2, 3.
double d[] = { 4, 5 };    // 2 double values: 4 and 5.
double d[10] = { 0 };     // 10 double values equal to 0.
```

Exercise 1

☐ Write a program that randomly initializes and shows the elements of an array with 10 elements.

```cpp
int main()
{
    const int SIZE = 10;
    int a[SIZE];              // Allocate space for 10 integers.
    for( int I = 0; I < SIZE; I++ )
            a[I] = rand(); // Assign a random integer number.

    // Print elements with their resulting value.
    for( int I = 0; I < SIZE; I++ )
            cout << I << "\t" << a[I] << endl;
}
```

Exercise 2

□ What do the following fragments of code do? Are they correct?

```cpp
int b[10] = { 0 }, I;
for( I = 0; I <= 10; I++ )
   b[I] = 1;

/********************************/

int size;
cin >> size;
int c[size] = { 1 };
for( int I = 0; I < size; I++ )
    c[I] = c[I] + c[I-1];
```

Exercise 3

- Find the first negative integer in an array of integers. The user should provide those values (up to 1000) and finish input when she introduces 0.

Arrays and Functions

- How do we declare that a function receives an array as argument?
    - returnType fName(arrayType aName[], int aSize);
- How do we pass an array to a function?
    - fName(argumentArray, argumentArraySize);
- Is it passed by reference or by value?
    - None of them: but think about it as by reference for now.

Exercise 4

☐ Write a function that finds the first negative number in an array of integers (see exercise 3), and returns 0 if no negative number was found.

```
int findFirstNegativeNumber( int a[], int size )
{
    for( int I = 0; I < size; I++ )
    {
            if( a[I] < 0 )
                    return a[I];
    }
    return 0;
}
```

Exercise 5

□ What does the following code do?

```cpp
int myFunction( int value )
{
    return value * value;
}

int main()
{
    int array[] = { 1, 2, 3 };
    for( int I = 0; I < 3; I++ )
            cout << myFunction( array[I] ) << endl;

    for( int I = 0; I < 3; I++ )
            cout << array[I] << endl;
}
```

Are these two outputs the same?

Exercise 6

□ What about this code?

```cpp
void myFunction( int& value )
{
    value *= value;
}

int main()
{
    int array[] = { 1, 2, 3 };
    for( int I = 0; I < 3; I++ )
    {
        myFunction( array[I] );
        cout << array[I] << endl;
    }

    for( int I = 0; I < 3; I++ )
        cout << array[I] << endl;
}
```

Are these two outputs the same?

const Qualifier

- When we pass arrays to functions we can modify their content, affecting thus the argument array we used when calling the function (*like pass by reference*).

- If we want to indicate the compiler that we will not change the contents of the array, we use **const** in front of the array type.

```
int findFirstNegativeNumber( const int a[], int size )
{
    int firstNegativeNumber = 0 ;
    for( int I = 0; I < size; I++ ) {
        if( a[I] < 0 )
        {
            firstNegativeNumber = a[I];
            break;
        }
    }
    return firstNegativeNumber;
}
```

Exercise 7

☐ Write a function that returns the position of the first element that is not smaller or equal than the one that follows it. Return -1 if there is no such element in the array.

```
int findFirstDisorder( const int a[], int n );
```

0	1	2	3	4	5	6	7	8	9
-1	2	3	7	1	3	4	0	1	9