Lecture 11.1

Topic

1. Arrays and Functions – Introduction

---

1. Arrays and Functions – Introduction

As an array is passed to a function, the array address is then sent to the function and it becomes available (or known) within the function. With this address, the function will be able to access the array elements.

> **When sending array arguments to functions, it is important to remember that the array itself (that means the values of array members/elements) will NOT be copied over to the function.**

The function will work through the array argument name in order to have access to the values as well as addresses of every member.

Because of having these addresses, inside the function, one may also be able to change the values of the array elements.

1.1 Passing Array Address

Assume that an integer array is to be passed to a function for printing. A possible prototype of such a function may be given as follows,

```
void printArrayVerionA( int arg[] );
```

In this prototype, an argument name is used with the square brackets to indicate an array. The type of the array can be any valid type.

The name can also be omitted in the prototype as below.

```
void printArrayVerionA( int [] );
```

Both of the above declarations say the following equivalent things:

- A value stored in an array variable is passed to the function, or
- An address of an array of integers is passed to the function.

For prototypes involved an array, there is no specific name of any array being mentioned or required.

Thus, to declare an array in a function prototype, one does not need to provide any argument name at all but just the square brackets (to refer to an “address”) should be sufficient.

1.2 Passing Array Size

Passing just the array address may not be enough for the function to process the array properly. An array argument must be declared with a size; and this size information should also be sent to the function as well.

Let's look at the prototypes below.

```
void printArrayVersionB( int iArray[], int iSize );
```

Or,

```
void printArrayVersionB( int[], int );
```

As the above the function was called, it has the all necessary information to work with the array. This means that the function has the location of the array (the array address) and the array size.

> **Note!**
>
> The array element values are accessible and these values may be changed by the function too.