# Lecture 06 Notes

## Derek Harter

CSci 515 Spring 2015 *<2015-02-18 Wed>*

# 1 First Session (11 - 11:40)

## 1.1 Introduction to Arrays

An array is a consecutive group of memory locations that all have the same type. When you declare an array, C will allocate a block of memory large enough to hold all of the values of the type you ask for. You access the elements of the array by using a subscript or index.

- Declaring an array

```
1  type arrayName[arraySize];
```

- Indexing an array

- Arrays in C are indexed **STARTING AT 0** (0 based indexing)

- The size of the array like `int c[10]` is 10 items, indexed from 0 to 9.

- When accessing all of the elements of an array, always use an indexed controlled for loop, and always index the loop from 0 up to N, like this:

```
1  int c[10];
2
3  for (int i = 0; i < 10; i++)
4  {
5  }
```

- It is good practice to declare a constant, and use it wherever you are referencing your array.

```
1 const int ARRAY_SIZE = 10;
2 int c[ARRAY_SIZE];
3
4 for (int i = 0; i < ARRAY_SIZE; i++)
5 {
6 }
```

- This way, if you need to change the size of the array and code processing the array elements, you change 1 single location, and all of the code will work correctly.

# 2 Second Session (11:45 - 12:30)

## 2.1 Examples of Using Arrays

- Initializing with a loop

- Specify array size with a defined constant (always do this).

- Summing the elements of an array

- Finding the minimum of the values of an array

- Finding the maximum of the values of an array

- Finding the average of the elements of an array

- Using elements of array as a counter

- Creating a histogram of elements, bucket size, etc.

# 3 Third Session (12:40 - 1:40)

## 3.1 Passing Arrays to Functions