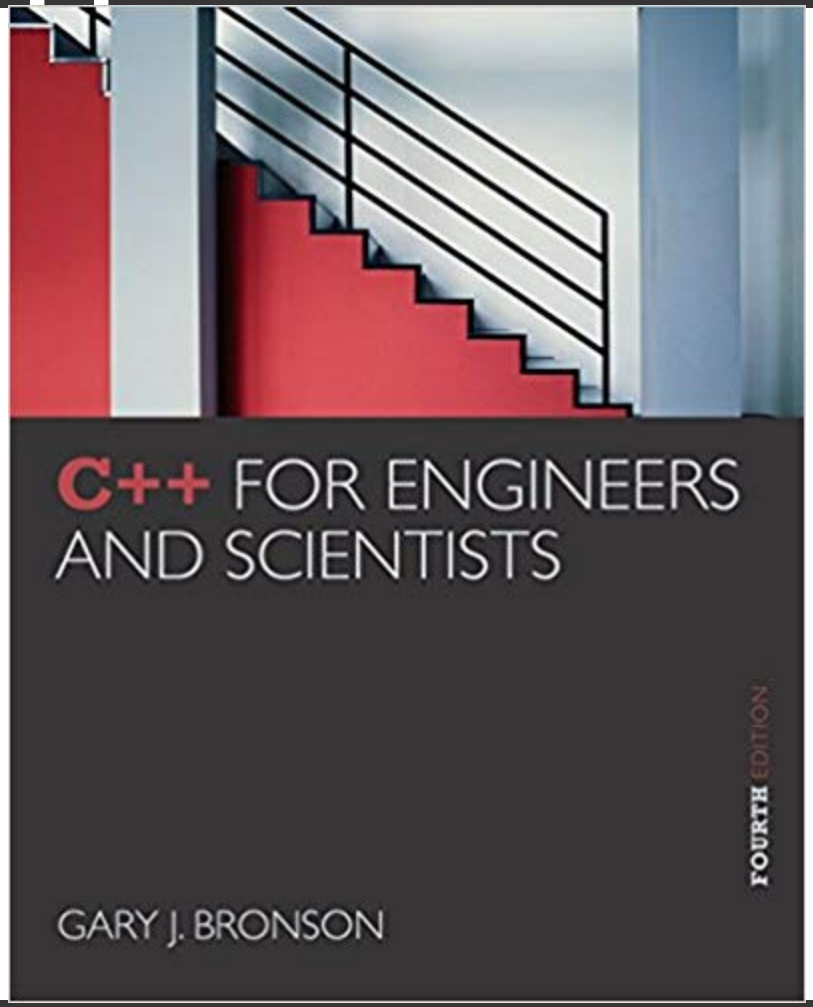
# ELEG 1043
# Computer Applications in Engineering



# Lab Course 5

# Acknowledgement


- Some of the slides or images are from various sources. The copyright of those materials belongs to their original owners.

# Chapter 7: Arrays

C++ FOR ENGINEERS AND SCIENTISTS

# Case Study

- Arrays are useful in applications that require multiple passes through the same set of data elements
  - Statistical Analysis
  - Array: *X* = [98, 82, 67, 54, 78, 83, 95, 76, 68, 63]
  - Calculating
    - Mean value
    - Standard Deviation

# Case Study

- Mean value

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N}$$

- Standard Deviation

$$\delta = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N-1}}$$

# Mean value

```
double findAvg(int nums[], int numel)
{
     int i;
    double sumnums = 0.0;
    for (i = 0; i < numel; i++)
          sumnums = sumnums + nums[i];
    return (sumnums / numel);
}
```

# Standard Deviation

```
double stdDev(int nums[], int numel, double avr)
{
    int i;
    double sumdevs = 0.0;
    for (i = 0; i < numel; i++)
        sumdevs = sumdevs + pow((nums[i] – avr),2);
    return (sqrt(sumdevs/(numel – 1.0)));
}
```

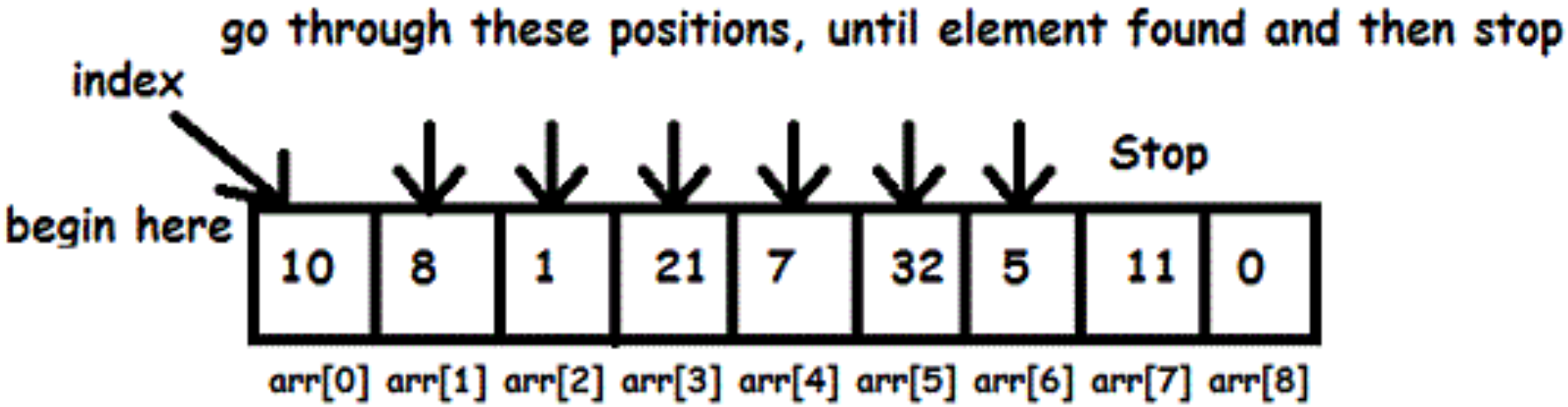
# Main Function

```
#include <iosteam>
using namespace std;
int main(){
    const int NUMELS = 10;
    int values[NUMELS] = {98, 82, 67, 54, 78, 83, 95, 76, 68, 63};
    double average, sDev;
    average = findAvg(values, NUMELS); // call the function
    sDev = stdDev(values, NUMELS, average); // call the function
    cout << "The average of the numbers is “<<average << endl;
    cout << "The standard deviation of the numbers is “<<sDev << endl;
    return 0;
}
```

# Linear Search

- Each item in the list is examined in the order in which it occurs
- Not a very efficient method for searching
- Advantage is that the list does not have to be in sorted order

# Linear Search (continued)




Element to search : 5

# Linear Search (continued)

```
/* Linear Search Function */
int linear_search(vector<int> v, int val)
{
    int key = -1;
    for (int i = 0; i < v.size(); i++)
    {
     if (v[i] == val)
     { key = i; break;}
    }
    return key;
}
```