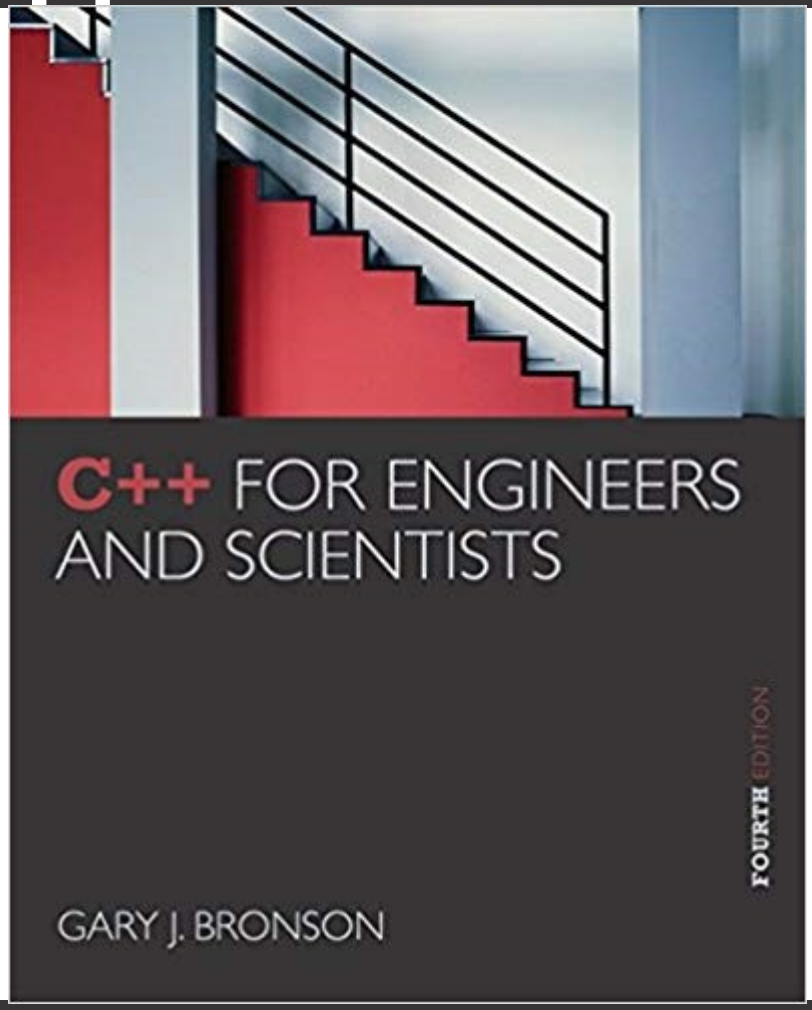
ELEG 1043
Computer Applications in Engineering

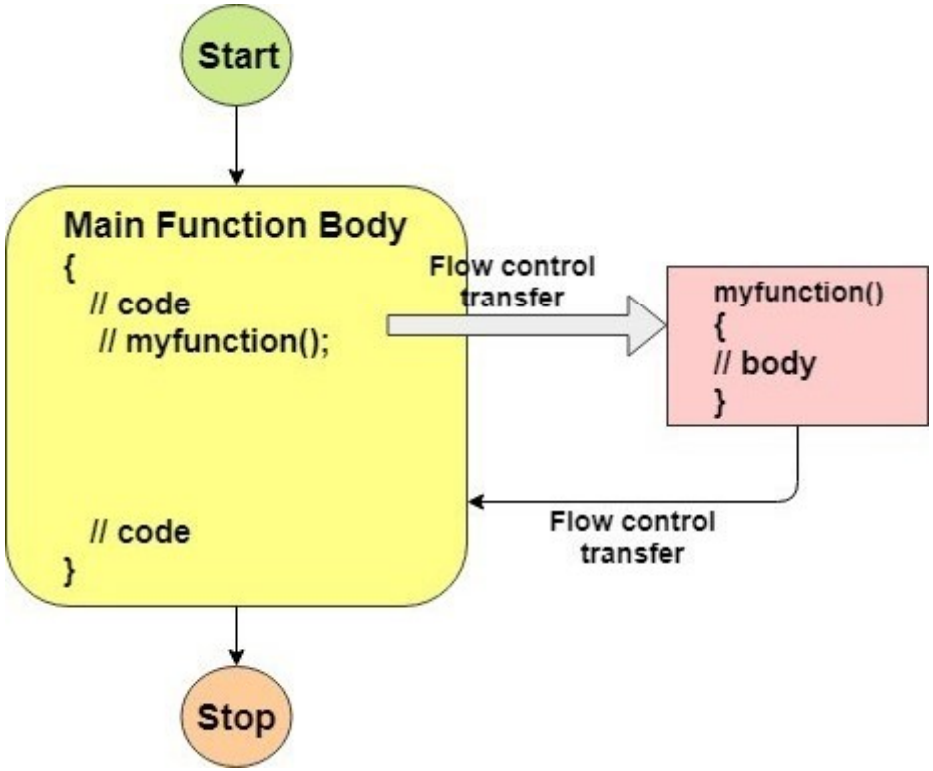


Lab Course 4

Acknowledgement

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

Function


Start
Main Function Body
{
// code
// myfunction();
// code
}
Flow control transfer
myfunction()
{
// body
}
Flow control transfer
Stop

Exercise 1

- **Write a function that is to add three numbers, and call this function in main function.**

Answer

```
#include <iostream>
using namespace std;
//Called Function
int add(int num1, int num2, int num3)
{
        int sum = num1 + num2 + num3;
        return sum;
}
//Calling Function
int main()
{
        int num1 = 1, num2 = 2, num3 = 5;
        int sum = add(num1, num2, num3);
        cout<<“the value is "<<sum<<endl;
        return 0;
}
```

Exercise 2

- **Write two functions with Function Overloading Technique. One function is to add two integer numbers, and the other is to add two double numbers.**

Answer

```
#include <iostream>
using namespace std;
int add(int num1, int num2){
        int value = num1 + num2;
        return value;
}
double add(double num1, double num2){
        double value = num1 + num2;
        return value;
}
int main(){
int num1 = 1, num2 = 2;
cout<<add(num1, num2);
double dnum1 = 0.1, dnum2 = 0.3;
cout<<add(dnum1, dnum2);
}
```

Exercise 3

- **Write a function that is to swap two numbers, and call this function in main function.**

Answer

```
#include <iostream>
using namespace std;
void swapnum(int &i, int &j) {
  int temp = i;
  i = j;
  j = temp;
}
int main(void) {
  int a = 10;
  int b = 20;
  cout<<"A is "<<a<<" and B is "<<b<<endl;
  swapnum(a, b);
  cout<<"After swapping two numbers"<<endl;
  cout<<"A is "<<a<<" and B is "<<b<<endl;
  return 0;
}
```

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)));
}
```

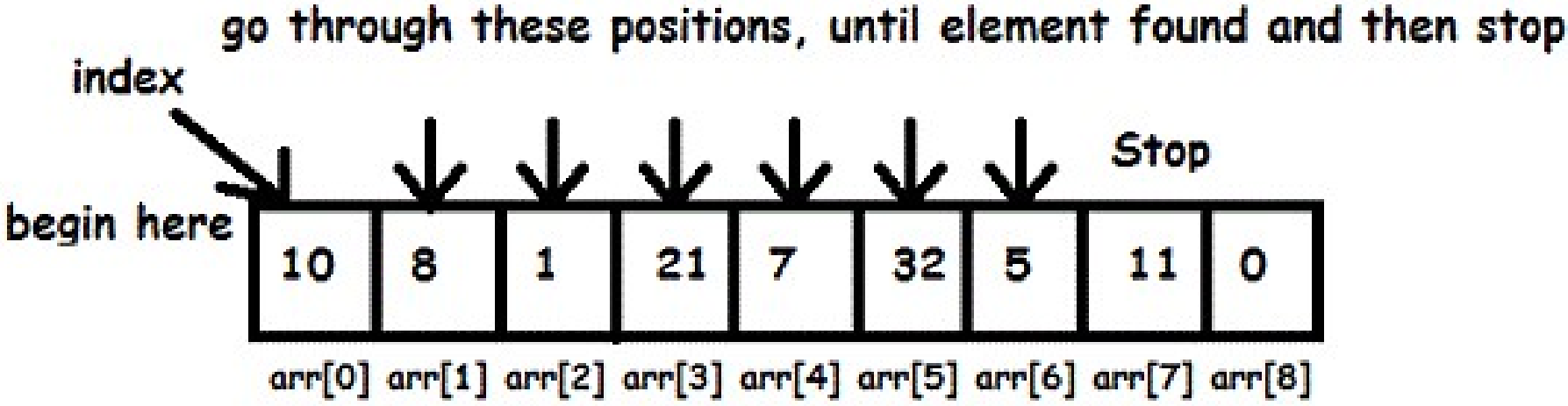
Main Function

```
#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;
}
```