

# PROGRAMMING LANGUAGES 1

Arrays & Strings



# AVERAGE CALCULATION

```
// Calculate the average of three numbers
int main()
{
    double x, y, z, average;
    cout << "Enter x, y, z: " << endl;
    cin >> x >> y >> z
    average = (x + y + z) / 3;
    cout<< "Average = " << average << endl;
    return 0;
}
```

# AVERAGE CALCULATION

```
// Now, calculate the average of 100 numbers
int main()
{
    double x[100];    // Array with 100 elements
    double average = 0;
    for(int i = 0; i < 100; i++) {
        cin >> x[i];    // Read into element i of the array
        average += x[i];
    }
    average /= 100;
    cout << "Average = " << average << endl;
}
```

# ARRAYS

- Use single variable to store a list of values
- Square brackets with index [] are used to access certain element
- Array size must be defined when defining the array
- Array size must be a **constant** value
- Array index starts at 0 up to  $n - 1$
- For loops are usually used to iterate over the array

# MATRICES

- Defined as 2-D arrays (array of arrays)

```
int x[2][3]; // Matrix with 2 rows and 3 columns
```

# ARRAY DEFINITION & INITIALIZATION

- ✓ `int x[5];`
- ✓ `const int n = 5; int x[n];`
- ✓ `int x[5] = {1,2,3,4,5};`
- ✓ `int x[] = {1,2,3,4,5};`
- ✗ `int x[];`
- ✗ `int n; int x[n];`
- ✓ `int x[2][3];`
- ✓ `int x[][3] = {{1,2,3}, {4,5,6}};`
- ✗ `int x[][] = {{1,2,3}, {4,5,6}};`
- ✗ `int x[][];`

# STRINGS

- C-style strings: valid in C and C++
- Standard Strings: valid in C++ only

# C-STYLE STRINGS

- Array of characters used to store strings (words or sentences)
- The last character of the array is the null character ‘\0’
- String size must be defined in advance

```
char message[] = "Hello";
```

```
// Equivalent to {'H' , 'e' , 'l' , 'l' , 'o' , '\0'}
```

```
char word[10];
```

```
strcpy(word, "Hello"); // Copy "Hello" to word
```

# C-STRING FUNCTIONS

- **strlen(s)** : Length of string s (without the null character)
- **strcpy(s1, s2)** : copy string s2 into s1
- **strcat(s1, s2)** : concatenate string s2 to the end of s1
- **strcmp(s1, s2)** : compare strings s1 and s2 (returns integer)

# C++ STANDARD STRING

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() {
```

```
    string message = "Hello";
```

```
    cout << "Length of message is " << message.length() << endl;
```

```
}
```

# C++ STRING OPERATIONS

- **s.length()** : Length of string s (without the null character)
- **s.c\_str()** : Convert to C-string (array character)
- **s1 = s2** : Copy s2 to s1
- **s1 += s2** : concatenate s2 to s1
- **s1 == s2** : check if s1 equals s2 (same for other relational operators)

# SUMMARY

- Arrays
- C-Strings
- C++ Standard Strings