

C++

PROGRAMMING LANGUAGE

L04-ARRAY+STRING

Mohammad Shaker

mohammadshaker.com

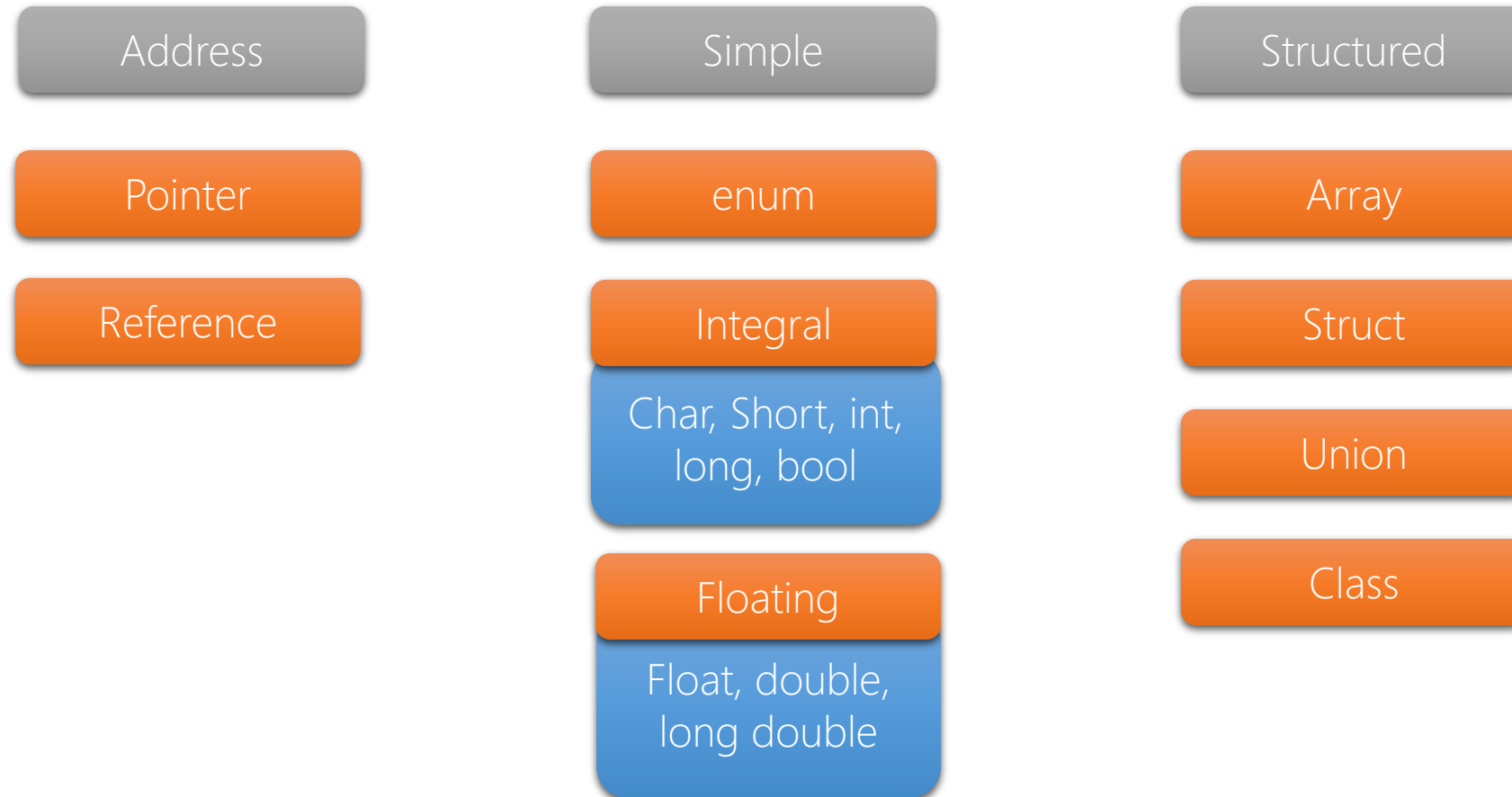
@ZGTRShaker

2010, 11, 12, 13, 14



Arrays

C++ data types



Arrays

- Note: n th element in position $n-1$



0 1 2 3 4 5 6 7



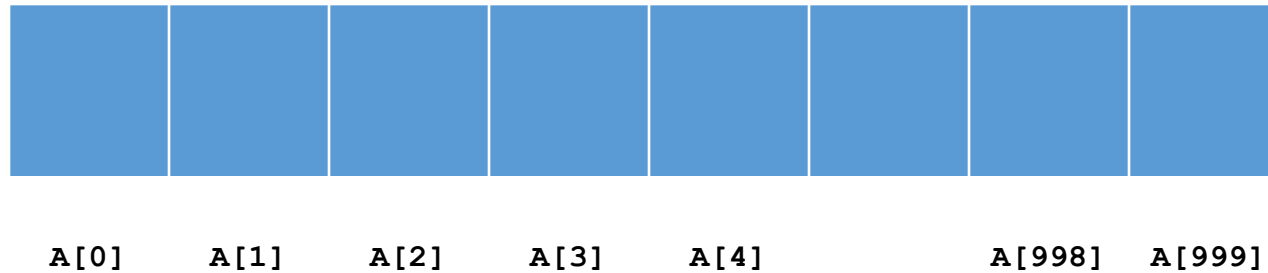
A[0] A[1] A[2] A[3] A[4] A[5] A[6] A[7]

Arrays

- Declaring arrays:

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [1000]; // means the positions are from 0 to 999
}
```



Arrays

- The array "variable" is the location of its first element in memory

```
#include <iostream>
using namespace::std;

void main(void)
{
    int intArr [10] = {4,5,56,6};
    cout << intArr << endl;
    system("pause");
}
```

```
0045FD08
Press any key to continue
```

When printing the array it print its location on memory by default

Just the char array
(char arr [])
Don't do that!
It prints the chars contained in it!
We'll see it in minutes;)

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int intArr [10] = {4,5,56,6};
    cout << intArr[0] << endl;
    system("pause");
}
```

```
4
Press any key to continue
```

When printing the array it print its location on memory by default

Just the **char** array
(**char arr []**)
Don't do that!
It prints the whole chars
contained in it!
We'll see it in minutes

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[3] = 0;
    A[0] = -1;
    for (int i = 0; i<=4; i ++)
    {
        cout << "A[" << i << "] = " << A[i] << endl;
    }
}
```

```
A[0] = -1
A[1] = 2092808
A[2] = 5040544
A[3] = 0
A[4] = 1523006256
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[0] = 1;
    A[1] = 3;
    A[2] = 4;
    A[3] = 7;
    A[4] = 89;
    for (int i = 0; i<=4; i ++)
    {
        cout << "A[" << i << "] = " << A[i] << endl;
    }
}
```

```
A[0] = 1
A[1] = 3
A[2] = 4
A[3] = 7
A[4] = 89
Press any key to continue
```


Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[0] = 1;
    A[1] = 3;
    A[2] = 4;
    A[3] = 7;
    A[4] = 89;
    A[5] = 34; //!!!!
    for (int i = 0; i<=4; i++)
    {
        cout << "A[" << i << "] = " << A[i] << endl;
    }
}
```

```
A[0] = 1
A[1] = 3
A[2] = 4
A[3] = 7
A[4] = 89
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[0] = 1;
    A[1] = 3;
    A[2] = 4;
    A[3] = 7;
    A[4] = 89;
    A[5] = 45;
    A[6] = 23;
    for (int i = 0; i<7; i++)
    {
        cout << "A[" << i << "] = " << A[i]
        << endl;
    }
}
```

```
c[0] = 1
c[1] = 3
c[2] = 4
c[3] = 7
c[4] = 89
c[5] = 5
c[6] = 23
```

And then a runtime error

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[3] = 0;
    A[0] = -1;
    for (int i = 0; i<=10; i ++)
    {
        cout << "A[" << i << "] = " << A[i] << endl;
    }
}
```

```
A[0] = -1
A[1] = 2092808
A[2] = 5040544
A[3] = 0
A[4] = 1523006256
A[5] = 5
A[6] = 2092672
A[7] = 7673537
A[8] = 5040544
A[9] = 2092736
A[10] = 1524674683
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[3] = 0;
    A[0] = 34;
    A[3] = 2;
    A[2] = 1;
    A[3+1] = 5;
    A[1] = 7;
    A[4] = 6;
    for (int i = 0; i<=4; i ++)
    {
        cout << "A[" << i << "] = " << A[i] << endl;
    }
}
```

```
A[0] = 34
A[1] = 7
A[2] = 1
A[3] = 2
A[4] = 6
Press any key to continue
```

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[3] = 0;
    A[0] = 34;
    A[3] = 2;
    A[2] = 1;
    A[3+1] = 5;
    A[1] = 7;
    A[4] = 6;
    for (int i = -1; i<=3; i ++)
    {
        cout << "A[" << i+1 << "] = " << A[i+1] << endl;
    }
}
```

```
A[0] = 34
A[1] = 7
A[2] = 1
A[3] = 2
A[4] = 6
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[3] = 0;
    A[0] = 34;
    A[3] = 2;
    A[2] = 1;
    A[3+1] = 5;
    A[1] = 7;
    A[4] = 6;
    for (int i = -1; i<=3; i ++)
    {
        cout << "A[" << i << "] = " << A[i] << endl;
    }
}
```

```
A[-1] = 1637993160
A[0] = 34
A[1] = 7
A[2] = 1
A[3] = 2
Press any key to continue
```

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A [5];
    A[3] = 0;
    A[0] = 34;
    A[3] = 2;
    A[2] = 1;
    A[3+1] = 5;
    A[1] = 7;
    A[4] = 6;
    for (int i = -1; i<4; i++)
    {
        cout << "A[" << i << "] = " << A[i] << endl;
    }
}
```

```
A[-1] = 1804389064
A[0] = 34
A[1] = 7
A[2] = 1
A[3] = 2
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int A      [5], X[2];
    A[3] = 0;
    A[0] = -1;
    X[0] = A[3];
    X[1] = A[2];
    for (int i = 0; i<2; i++)
    {
        cout << "X[" << i << "] = " << X[i] << endl;
    }
}
```

```
X[0] = 0
X[1] = 3011612
Press any key to continue
```

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[5] = {3,5,2,4,33};    // initializing
    for (int i = 0; i<5; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}
```

```
a[0] = 3
a[1] = 5
a[2] = 2
a[3] = 4
a[4] = 33
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[5] = {3,5,2,4,33,12}; // initializing
    for (int i = 0; i<5; i++)
    {
        cout << "a[" << i << "] = " << a[i]
    << endl;
    }
}
```

Compiler error

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[5] = {3,5,2}; // initializing
    for (int i = 0; i<5; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}
```

```
a[0] = 3
a[1] = 5
a[2] = 2
a[3] = 0
a[4] = 0
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[5] = {4}; // initializing
    for (int i = 0; i<5; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}
```

```
a[0] = 4
a[1] = 0
a[2] = 0
a[3] = 0
a[4] = 0
Press any key to continue
```

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[5] = {0};    // initializing
    for (int i = 0; i<5; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}
```

```
a[0] = 0
a[1] = 0
a[2] = 0
a[3] = 0
a[4] = 0
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[] = {32,34,12,23};
    for (int i = 0; i<4; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}
```

```
a[0] = 32
a[1] = 34
a[2] = 12
a[3] = 23
Press any key to continue
```

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[] = {32,34,12,23};
    for (int i = 0; i<5; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}
```

```
a[0] = 32
a[1] = 34
a[2] = 12
a[3] = 23
a[4] = 434378978
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int a[5];
    for (int i = 0; i<5; i++)
    {
        a[i] = i+2;
    }
    cout << "_____" << endl;
    cout << "The output array is: " << endl;
    for (int i=0; i<5; i++)
    {
        cout << a[i] << endl;
    }
}
```

```
_____  
The output array is:  
2  
3  
4  
5  
6
```


Arrays

```
#include <iostream>
using namespace::std;

const int ArrSize = 4;

void main(void)
{
    int a[ArrSize];
    for(int i = 0; i<ArrSize; i++)
    {
        a[i]=i*ArrSize;
    }
    cout << a[3];
}
```

12

```
#include <iostream>
using namespace::std;

const int ArrSize = 4;

void main(void)
{
    ArrSize = 7;
    int a[ArrSize];
    for(int i = 0; i<ArrSize; i++)
    {
        a[i]=i*ArrSize;
    }
    cout << a[3];
}
```

Compiler error

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    float a[4] = {30};
    cout << a[0] << endl;
}
```

30

```
#include <iostream>
using namespace::std;

void main(void)
{
    int foo = 4;
    float a[foo] = {30};
    cout << a[0] << endl;
}
```

Compile error. Array length should be constant

Arrays

```
#include <iostream>
using namespace::std;

const int foo = 0;

void main(void)
{
    float a[foo] = {30};
    cout << a[0] << endl;
}
```

Compiler error, can't allocate any array of size Zero

```
#include <iostream>
using namespace::std;

void main(void)
{
    float a[4] = {30., 32, 33.4, 23.0 };
    cout << a[0] << endl;
}
```

30

But with warning of truncation from double to float

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int x = 44.;
}
```

Compile & Run

But with warning for converting from **double** to int. Possible loss of data

```
#include <iostream>
using namespace::std;

void main(void)
{
    float a[4] = {30., 32, 33.4, 23.0 };
    int x[2] = {34,0};
    a[x[1]]=3;
    for (int i = 0; i < 4; i++)
        cout << a[i] << endl;
}
```

```
3
32
33.4
23
Press any key to continue
```

Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    float a[4] = {30., 32, 33.4, 23.0 };
    float x[2] = {34,0};
    a[x[1]]=3;
    for (int i = 0; i < 4; i++)
        cout << a[i] << endl;
}
```

Compiler error. Index in not of an integral type

```
#include <iostream>
using namespace::std;

void PassingArr(int a [])
{
    cout << a[0] << endl;
}

void main(void)
{
    int Arr [] = {3,2};
    PassingArr(Arr);
}
```

3

Arrays

```
#include <iostream>
using namespace::std;

void PassingArr(int a [])
{
    cout << a[0] << endl;
}

void main(void)
{
    int Arr [] = {3,2};
    PassingArr(Arr[]);
}
```

Compiler error Passing the array with subscript []

```
#include <iostream>
using namespace::std;

void PassingArr(int a [])
{
    cout << a[0] << endl;
}

void main(void)
{
    int Arr [] = {3,2};
    PassingArr(Arr);
}
```

3

Arrays

```
#include <iostream>
using namespace::std;

void PassingArrVal(int a[])
{
    a[0]+=2;
}

void main(void)
{
    int Arr [] = {3,2};
    PassingArrVal(Arr);
    cout << Arr[0] << endl;
    PassingArrVal(Arr);
    cout << Arr[0] << endl;
}
```

5
7

Passing arrays by value is passing it with reference not by value!
Why?

```
#include <iostream>
using namespace::std;

void PassingArrVal(int a[])
{
    a[0]+=2;
}
void PassingArrRef1(int &a[])
{
    a[0]+=2;
}
void main(void)
{
    int Arr [] = {3,2};
    PassingArrVal(Arr);
    cout << Arr[0] << endl;
    PassingArrRef(Arr);
    cout << Arr[0] << endl;
}
```

Compiler error. Can't pass arrays with values. it's passed by reference.

Arrays

```
#include <iostream>
using namespace::std;

void PassingArr(const int a [])
{
    cout << a[0] << endl;
}

void main(void)
{
    int Arr[2] = {1,3};
    PassingArr(Arr);
}
```

1

```
#include <iostream>
using namespace::std;

void PassingArr(const int a [])
{
    a[0]+=2;
}

void main(void)
{
    int Arr [2];
    PassingArr(Arr);
    cout << Arr[0] << endl;
}
```

Compiler error, can't modify const arrays



Multiple Subscripted Arrays

Multiple Subscripted Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int Arr[3][4];    // 3 rows, 4 columns
}
```

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Array name

Row subscript

Column subscript

Multiple Subscripted Arrays

```
#include <iostream>
using namespace::std;
```

Compiler error Should be [][]

```
void main(void)
{
    int Arr[2,3];
}
```

```
#include <iostream>
using namespace::std;
```

Compiler error, missing
subscripts size

```
void main(void)
{
    int Arr [][];
}
```

```
#include <iostream>
using namespace::std;
```

Compiler error, missing
subscript size

```
void main(void)
{
    int Arr [][][3];
}
```

Multiple Subscripted Arrays

```
int Arr[2][3] = { {2,3,5}, {4,1,3} };  
                //  row#1   row#2
```

2	3	5
4	1	3

```
int Arr[2][3] = { {2,3}, {4,1,3} };  
                //  row#1   row#2
```

2	3	0
4	1	3

```
int Arr[2][3] = { {2}, {4,3} };
```

2	0	0
4	3	0

Multiple Subscripted Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int Arr[2][3] = { {2,3,5}, {4,3} };
    cout << Arr[2][3] << endl;
}
```

246547657

Not a Compiler error, but the stored value in that location in memory

```
#include <iostream>
using namespace::std;

void main(void)
{
    int Arr[2][3] = { {2,3,5}, {4,3} };
    cout << Arr[1][2] << endl;
}
```

0



Passing Arrays to Functions

Passing Arrays to Functions

```
#include <iostream>
using namespace::std;

void ArrFun(int A [2][3])
{
    cout << A [1][1];
}

void main(void)
{
    int Arr[2][3] = { {2,3,5}, {4,3} };
    ArrFun(Arr);
}
```

Passing Arrays to Functions

```
#include <iostream>
using namespace::std;

void ArrFun(int A [2][3])
{
    cout << A [1][1];
}

void main(void)
{
    int Arr[2][3] = { {2,3,5}, {4,3} };
    ArrFun(Arr[2][3]);
}
```

Compiler error, [2][3]

```
#include <iostream>
using namespace::std;

void ArrFun(int A [][3])
{
    cout << A [1][1];
}

void main(void)
{
    int Arr[2][3] = { {2,3,5}, {4,3} };
    ArrFun(Arr);
}
```

3

No compiler error for missing the first subscript in the function's prototype
like single-scripted arrays

Passing Arrays to Functions

```
#include <iostream>
using namespace::std;

void ArrFun(int A [][])
{
    cout << A [1][1];
}

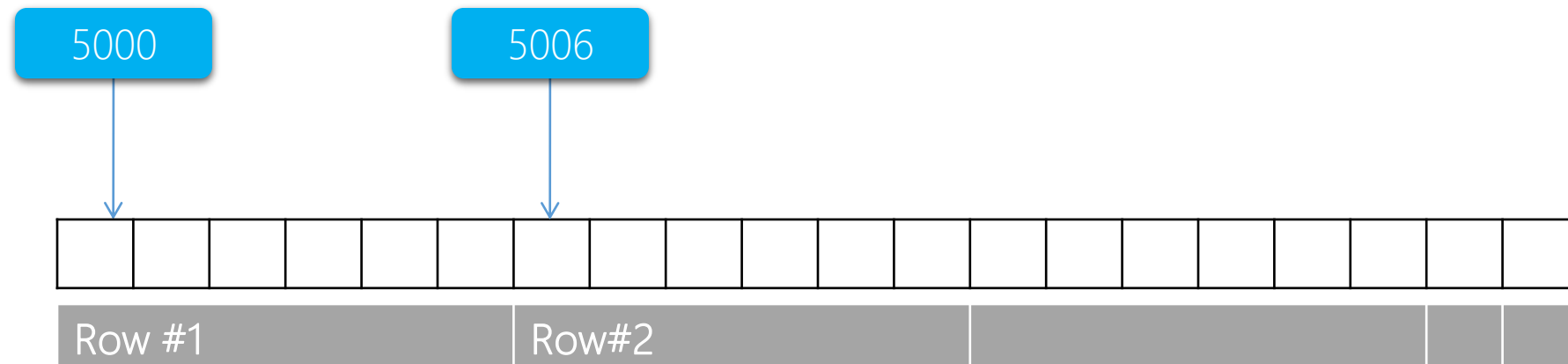
void main(void)
{
    int Arr[2][3] = { {2,3,5}, {4,3} };
    ArrFun(Arr);
}
```

Compiler error, missing the second subscript in the function's prototype

Passing Arrays to Functions

```
#include <iostream>
using namespace::std;
const int Rows = 2;
const int Columns = 3;

void main(void)
{
    int Arr[Rows][Columns];
}
```





Fundamentals of Strings

String data type

- String
 - Sequence of Zero or more character
 - The position of the first character is 0 and not 1!
 - Length of the string is the number of characters in it.
 - Represented by enclosed with double quote marks " "
 - (we will see it later)

String data type

```
#include<iostream>
using namespace::std;

void main()
{
    string s = "Hello!";
    cout << s;
    system("pause");
}
```

Compiler error! See how to deal with it later

cout deals with numbers (int, bool, char) only!



String as Array of Characters

Strings using Arrays

- String:
 - It's an array of characters
- All strings end with null ('\0')

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [] = "Hello";
    // null character implicitly added at the end
    char str2 [] = {'H','e','l','l','o','\0'};
    // null character explicitly added
}
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [] = "hello";
    char str2 [] = {'h','e','l','l','o','\0'};

    for (int i = 0; i < 6; i++)
    {
        cout << int(str1[i])
        << endl;
    }

    cout << "-----" << endl;

    for (int i = 0; i < 6; i++)
    {
        cout << int(str2[i])
        << endl;
    }
}
```

```
104
101
108
108
111
0
-----
104
101
108
108
111
0
```


Strings using Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [] = "hello";
    char str2 [] = {'h','e','l','l','o','\0'};
    (str1==str2)? cout << "Yes!": cout << "No!";
}
```

No!

Coz we are comparing references and not values!
Remember it's an **array** of chars.

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [] = "hello";
    char str2 [] = {'h','e','l','l','o','\0'};
    (str1[0]==str2[0])? cout << "Yes!":
        cout << "No!";
}
```

Yes!

Coz we are comparing values and not references!

Strings using Arrays

```
#include <iostream>
using namespace::std;
```

```
void main(void)
```

```
{
    char str1 [] = "hello";
    char str2 [] = {'h','e','l','l','o','\0'};
    (str1==str2)? cout << "Yes!": cout << "No!";
    cout << "\n";

    for (int i = 0; i < 6; i++)
    {
        cout << int(str1[i]) << endl;
    }

    cout << "-----" << endl;

    for (int i = 0; i < 6; i++)
    {
        cout << int(str2[i]) << endl;
    }
}
```

```
No!
104
101
108
108
111
111
0
-----
104
101
108
108
111
111
0
```

```
#include <iostream>
using namespace::std;
```

```
void main(void)
```

```
{
    char str1 [] = "hello";
    char str2 [] = {'h','e','l','l','o','\0'};
    (str1==str2)? cout << "Yes!": cout << "No!";
    cout << "\n";

    for (int i = 0; i < 6; i++)
    {
        cout << int(str1[i]) << endl;
    }

    cout << "-----" << endl;

    for (int i = 0; i < 6; i++)
    {
        cout << int(str2[i]) << endl;
    }
}
```

```
No!
104
101
108
108
111
111
0
-----
104
101
108
108
111
111
0
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [10];
    cin >> str1;
    cout << "The input string as a whole!, is " << str1 << endl;
}
```

Only string array can be printed by calling it by its variable name. Other type of array will simply print their location in memory

```
Hi, My name is Mohammad
The input string as a whole!, is Hi, My name is Mohammad
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    int intArr [10] = {0,1,1,12,34};

    cout << "The input string as a whole!, is " << intArr << endl;
    system("pause");
}
```

```
The input string as a whole!, is 0043F958
Press any key to continue
```

Strings using Arrays

```
#include <iostream>
using namespace::std;
```

```
void main(void)
```

```
{
    char str1 [] = "hello";
    char str2 [] = {'h','e','l','l','o','\0'};
    (str1==str2)? cout << "Yes!": cout << "No!";
    cout << "\n";

    for (int i = 0; i < 6; i++)
    {
        cout << str1[i];
    }

    cout << "\n-----" << endl;

    for (int i = 0; i < 6; i++)
    {
        cout <<str2[i];
    }
    cout << endl;
}
```

```
No!
hello
-----
hello
```

```
#include <iostream>
using namespace::std;
```

```
void main(void)
```

```
{
    char str1 [] = "hello";
    char str2 [] = {'h','e','l','l','o','\0'};
    (str1[0]==str2[0])? cout << "Yes!":
        cout << "No!";
}
```

```
Yes!
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [10];
    cin >> str1;
    cout << "The input string as a whole!, is "
    << str1 << endl;
}
```

```
ZeeeeeeeeeeeeeeZeeeeeeeeeeeee
The input string as a whole!, is ZeeeeeeeeeeeeeeZeeeeeeeeeeeee
Press any key to continue
```

Print but Runtime exception! Out of bound exception!

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [10];
    cin >> str1;    // WeWe
    cout << "The input string as a whole!, is "
    << str1 << endl;
}
```

```
WeWe sd
The input string as a whole!, is WeWe
Press any key to continue
```

Stops at the first white space

Strings using Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [10];
    cin >> str1;
    cout << "The input string as a whole!, is "
    << str1 << endl;
}
```

```
ZeeeeeeeeeeeeeeeeZeeeeeeeeeeeeee
The input string as a whole!, is ZeeeeeeeeeeeeeeeeZeeeeeeeeeeeeee
Press any key to continue
```

```
#include <iostream>
using namespace::std;

void main(void)
{
    char str1 [10];
    cin >> str1;    // WeWe
    cout << "The input string as a whole!, is "
    << str1 << endl;
}
```

```
WeWe sd
The input string as a whole!, is WeWe
Press any key to continue
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void main(void)
{
    int str1 [10];
    cin >> str1;

    cout << "The input string as a whole!, is " << str1 << endl;
    system("pause");
}
```

Compiler error (`cin >> str1;`), it's not a char array!
U can't input it as a whole!

```
#include <iostream>
using namespace::std;

void main(void)
{
    int intArr [10] = {0,1,1,12,34};

    cout << "The input string as a whole!, is " << intArr << endl;
    system("pause");
}
```

The input string as a whole!, is 0043F958
Press any key to continue

Strings using Arrays

```
#include <iostream>
using namespace::std;

void AutoArr()
{
    int Arr1 [3] = {1,2,3};
    cout << "Auto Arr" << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i
            << "]" << " = " << Arr1[i] << endl;
    }
}

void main(void)
{
    AutoArr();
}
```

```
Auto Arr
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
```

```
#include <iostream>
using namespace::std;

void AutoArr()
{
    int Arr1 [3] = {1,2,3};
    cout << "Auto Arr" << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]" << " = " <<
Arr1[i] << endl;
    }
}

void main(void)
{
    AutoArr;
}
```

```
Press any key to continue
```


Strings using Arrays

```
#include <iostream>
using namespace::std;

void AutoArr()
{
    int Arr1 [3] = {1,2,3};
    cout << "Auto Arr: " << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= "
              << Arr1[i]<< endl;
    }
    cout << "Auto Arr after +2 to all elemets:"
          <<endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " <<
              (Arr1[i]=Arr1[i]+2) << endl;
    }
}

void main(void)
{
    AutoArr();
}
```

```
Auto Arr:
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
Auto Arr after +2 to all elemets:
Arr1[0]=3
Arr1[1]=4
Arr1[2]=5
Press any key to continue
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void AutoArr()
{
    int Arr1 [3] = {1,2,3};
    cout << "Auto Arr: " << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= "
              << Arr1[i]<< endl;
    }
    cout << "Auto Arr after +2 to all elemets:"
          <<endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " <<
              (Arr1[i]+=2) << endl;
    }
}

void main(void)
{
    AutoArr();
}
```

```
Auto Arr:
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
Auto Arr after +2 to all elemets:
Arr1[0]=3
Arr1[1]=4
Arr1[2]=5
Press any key to continue
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void AutoArr()
{
    int Arr1 [3] = {1,2,3};
    cout << "Auto Arr: " << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= "
              << Arr1[i]<< endl;
    }
    cout << "Auto Arr after +2 to all elemets:"
          <<endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " <<
              (Arr1[i]+=2) << endl;
    }
}

void main(void)
{
    AutoArr();
    AutoArr();
}
```

```
Auto Arr:
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
Auto Arr after +2 to all elemets:
Arr1[0]=3
Arr1[1]=4
Arr1[2]=5
Auto Arr:
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
Auto Arr after +2 to all elemets:
Arr1[0]=3
Arr1[1]=4
Arr1[2]=5
Press any key to continue
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void AutoArr()
{
    int Arr1 [3] = {1,2,3};
    cout << "Auto Arr: " << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= "
              << Arr1[i]<< endl;
    }
    cout << "Auto Arr after +2 to all elemets:"
          <<endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " <<
              (Arr1[i]=+2) << endl;
    }
}

void main(void)
{
    AutoArr();
}
```

```
Auto Arr:
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
Auto Arr after +2 to all elemets:
Arr1[0]=2
Arr1[1]=2
Arr1[2]=2
Press any key to continue
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void StaticArr()
{
    static int Arr1 [3] = {1,2,3};
    cout << "Arr: " << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " << Arr1[i] << endl;
    }
    cout << "Arr after +2 to all elemets:" <<endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " << (Arr1[i]+=2)
        <<endl;
    }
}

void main(void)
{
    StaticArr();
    StaticArr();
}
```

```
Arr:
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
Arr after +2 to all elemets:
Arr1[0]=3
Arr1[1]=4
Arr1[2]=5
Arr:
Arr1[0]=3
Arr1[1]=4
Arr1[2]=5
Arr after +2 to all elemets:
Arr1[0]=5
Arr1[1]=6
Arr1[2]=7
Press any key to continue
```

Strings using Arrays

```
#include <iostream>
using namespace::std;

void StaticArr()
{
    static int Arr1 [3] = {1,2,3};
    cout << "Arr: " << endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " << Arr1[i] << endl;
    }
    cout << "Arr after +2 to all elemets:" <<endl;
    for (int i=0; i<3; i++)
    {
        cout << "Arr1[" << i << "]= " << (Arr1[i]=+2)
        <<endl;
    }
}

void main(void)
{
    StaticArr();
    StaticArr();
}
```

```
Arr:
Arr1[0]=1
Arr1[1]=2
Arr1[2]=3
Arr after +2 to all elemets:
Arr1[0]=2
Arr1[1]=2
Arr1[2]=2
Arr:
Arr1[0]=2
Arr1[1]=2
Arr1[2]=2
Arr after +2 to all elemets:
Arr1[0]=2
Arr1[1]=2
Arr1[2]=2
Press any key to continue
```