String in C++

M.Sc. Dina Rafaa

string in C++

C++ has TWO different ways of storing and manipulating strings:

- character strings (c-strings)
- strings Class

The char Data Type

The char data type is used to store individual characters. A variable of the char data type can hold only one character at a time.

- char ch;

In C++, *character literals* are enclosed in single quotation marks.

- ch = ‘g’;
- ch = “g”; // Error! char variables can hold only one character.

The char Data Type (continued)

Example: works with char.

```
#include<iostream>
using namespace std;
void main()
{
char ch;
ch = 'A';
cout << ch << endl;
ch = 'B';
cout << ch << endl;
system("pause");
}
```

Program output:

A
B

The char Data Type (continued)

When the character is stored in memory it is actually the numeric code that is stored. When the computer is instructed to print the value on the screen, it displays the character that corresponds with the numeric code. The ASCII character set shown below.

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	¶	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	¦
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

The char Data Type (continued)

Example: demonstrates the close relationship between the characters and the integers.

```
#include<iostream>
using namespace std;
void main()
{
char ch;
ch = 65;
cout << ch << endl;
ch = 66;
cout << ch << endl;
system("pause");
}
```

Program output:

A
B

character strings (c-string)

Character strings (c-strings): using one-dimensional array of characters that is terminated by a null character '\0'.

- char st[5] = “abcd”;

This array is capable of storing maximum of 4 characters. One character is reserved for storing ‘\0’. Thus the number of elements that can be stored is always (the size of the array-1).

- char st[20] = “Hello There”;

H	e	l	l	o		W	o	r	l	d	\0

The above string will have 11 characters and 1 byte for the ‘\0’. Thus size of the array st will be 12.

character strings (c-string) (continued)

We can initialize an array of characters in the following way also:

- `char st[ ] = “abcd”;`

In this case the array is initialized to the mentioned string and the size of the array is the number of the character in the string literal and the null character is automatically inserted at the end of the string. We can also write:

- `char st[] = {'a', 'b', 'c', 'd', '\0’};`

Here the array of characters is initialized character by character and the ‘\o’ has to be inserted at the end by the programmer.

character strings (c-string) (continued)

Some more examples of initialization:

- char animal[ ] = “Lion”;

L	i	o	n	\0

- char location[ ] = “New York”;

N	e	w		Y	o	r	k	\0

- char serial_no[ ] = “A011”;

A	0	1	1	\0

- char company[ ] = “DELL”;

D	E	L	L	\0

strings Class (continued)

Example: displays a string stored in a char array.

```
#include<iostream>
using namespace std;
void main()
{
char line1[8],line2[]="Hello World";
cin >> line1;
cout << "The 1st line is : " << line1 << endl << "The 2nd line is : " << line2 << endl;
system("pause");
}
```

Program output:

1234567
The 1st line is : 1234567
The 2nd line is : Hello World

strings Class

C++ provides something called the string class that allows the programmer to create a string type variable. The first step to use the string class is writing the string header file:

- # include <string>

The next step is to define a string type variable:

- string st ;

Finally assign the defined string st with the assignment operator:

- st = “Hello World!”;

strings Class (continued)

Example: demonstrates the string class.

```
#include<iostream>
#include<string>     // required for the string class.
using namespace std;
void main()
{
string movieTitle;
movieTitle = "Wheels of Fury";
cout << "My favorite movie is " << movieTitle << endl;
system("pause");
}
```

Program output:

My favorite movie is Wheels of Fury

Library Functions for working with strings

The c++ library has numerous functions for handling strings. These functions perform various tests and manipulations.

I. The Length of the string:

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
void main()
{
string st = "Thomas Edison";
int length = st.length();
cout << "the string length is : " << length << endl;
system("pause");
}
```

Library Functions for working with strings (continued)

I. The Length of the string:

B. With string class:

```
#include<iostream>
#include<string>     // required for the string class.
using namespace std;
void main()
{
string st = "Thomas Edison";
int length = st.length();
cout << "the string length is : " << length << endl;
system("pause");
}
```

Library Functions for working with strings (continued)

2. Concatenates one string to another:

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
char ch1[50] = "Learning C++ is fun",ch2[50] = " and easy";
strcat(ch1, ch2);
cout << ch1;
return 0;
}
```

Library Functions for working with strings (continued)

2. Concatenates one string to another:

B. With string class:

```
#include<iostream>
#include<string>     // required for the string class.
using namespace std;
void main()
{
string st1 = "Thomas“, st2 = “Edison“;
cout << "the string length is : " << st1+” ”+st2 << endl;
system("pause");
}
```

Library Functions for working with strings (continued)

3. Copy one string to another:

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
char ch1[50] = "Hello", ch2[50] = "World!";
strcpy(ch1, ch2);
cout << ch1;
return 0;
}
```

Library Functions for working with strings (continued)

3. Copy one string to another:

B. With string class:

```
#include<iostream>
#include<string>     // required for the string class.
using namespace std;
void main()
{
string st1 = "Hello", st2 = "World!";
st1=st2;
cout << st1 << endl << st2 << endl;
system("pause");
}
```

Library Functions for working with strings (continued)

4. Read a string:

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
char ch1[50];
cin.getline(ch1);
cout << ch1;
return 0;
}
```

Library Functions for working with strings (continued)

4. Read a string:

B. With string class:

```
#include<iostream>
#include<string>     // required for the string class.
using namespace std;
void main()
{
string st1, st2 = "World!";
getline(cin, st1);
cout << st1 << endl << st2 << endl;
st1=st2;
cout << st1 << endl << st2 << endl;
system("pause");
}
```

Library Functions for working with strings (continued)

5. Compare two strings:

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
char ch1[10] = “Hello", ch2[10] = "World!";
if(strcmp(ch1,ch2)==0)
cout << ch1 << “ & " << ch2 << " are Equal" << endl;
else
cout << ch1 << “ & " << ch2 << " are not Equal" << endl;
return 0;
}
```

Library Functions for working with strings (continued)

5. Compare two strings:

B. With string class:

```
#include<iostream>
#include<string>     // required for the string class.
using namespace std;
void main()
{
string st1 = "Hello", st2 = "World!";
if(st1 == st2)
cout << st1 << "&" << st2 << " are Equal" << endl;
else
cout << st1 << "&" << st2 << " are not Equal" << endl;
system("pause");
}
```

Thank You …