



String in C++ (2)

M.Sc. Dina Radaa

Library Functions for working with **strings**

6. Searches for a substring inside of a string:

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
void main()
{
    char ch[50] = "Learning C++ is fun";
    char *p = nullptr;
    p = strstr(ch, "fun");

    if (p != nullptr) OR (????????)
        cout << "Matching product was found = " << p << endl;
    else
        cout << "No matching product was found" << endl;
    system("pause");
}
```

Library Functions for working with **strings** (continued)

6. Searches for a substring inside of a string:

B. With string class:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    string st1 = "Learning C++ is fun", st2 = "fun";
    int p = st1.find(st2);

    if ( p <= st1.length())    OR (?????????)
        cout << "Matching product was found"<< endl;
    else
        cout << "No matching product was found" << endl;
    system("pause");
}
```

How you can find the range of the substring?

Library Functions for working with **strings** (continued)

7. **Sorting strings alphabetically:**

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
void main()
{
    char n1[30], n2[30];
    cout << "Enter the first name: ";
    cin >> n1;

    cout << "Enter the second name: ";
    cin >> n2;

    if (strcmp(n1,n2) < 0)
        cout << n1 << endl << n2 << endl;
    else if(strcmp(n1, n2) > 0)
        cout << n2 << endl << n1 << endl;
    else
        cout << "You entered the same name twice" << endl;
    system("pause");
}
```

OR (???????)

Library Functions for working with **strings** (continued)

7. **Sorting strings alphabetically:**

B. With string class:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    string n1, n2;
    cout << "Enter the first name: ";
    cin >> n1;
    cout << "Enter the second name: ";
    cin >> n2;
    if (n1 < n2)
        cout << n1 << endl << n2 << endl;
    else if (n1 > n2)
        cout << n2 << endl << n1 << endl;
    else
        cout << "You entered the same name twice" << endl;
    system("pause");
}
```

} **OR (???????)**

Library Functions for working with **strings** (continued)

8. Using ! with comparison process:

A. With c-string:

```
#include<iostream>
#include<cstring>
using namespace std;
void main()
{
    char n1[30], n2[30];
    cout << "Enter the first name: ";
    cin >> n1;
    cout << "Enter the second name: ";
    cin >> n2;
    if (! strcmp(n1,n2) )
        cout << "You entered the same name twice" << endl;
    else
        cout << "You entered different names" << endl;
    system("pause");
}
```

Library Functions for working with **strings** (continued)

8. Using ! with comparison process:

B. With string class:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    string n1, n2;
    cout << "Enter the first name: ";
    cin >> n1;
    cout << "Enter the second name: ";
    cin >> n2;
    if (n1 != n2 )
        cout << "You entered different names" << endl;
    else
        cout << "You entered the same name twice" << endl;
    system("pause");
}
```

Library Functions for working with **strings** (continued)

9. Convert string Class into c-string:

We can convert the contents of a string class to a c-string by writing the following code:

```
#include<iostream>
#include<string>
#include<cstdio>
using namespace std;
int main()
{
    string st;
    cout << "Enter your string class = ";
    cin >> st;
    char ch[50];
    strcpy_s(ch,st.c_str());
    cout << "Your c-string is : " << ch << endl;
    return 0;
}
```

Find the length of the c-string (ch)



Library Functions for working with **strings** (continued)

10. Clean the string:

We can clean the string by deleting all the characters stored in it:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    string st;
    cout << "Enter your string: ";
    cin >> st;
    st.clear();
    if (st == "")
        cout << "Is Empty " << "(" << st << ")";
    else
        cout << "Is NOT Empty " << "(" << st << ")";
    system("pause");
}
```



Library Functions for working with **strings** (continued)

II. Check the string value:

We can check the string value as the following:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    string st;
    cout << "Enter your string: ";
    cin >> st;
    st.clear();
    if (st.empty())
        cout << "Is Empty " << "(" << st << ")";
    else
        cout << "Is NOT Empty " << "(" << st << ")";
    system("pause");
}
```



Library Functions for working with **strings** (continued)

12. Return character in the string:

We can return a specific character in the string as the following:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    string st;
    cout << "Enter your string: ";
    cin >> st;
    cout << st.at(2) << endl;
    system("pause");
}
```

Library Functions for working with **strings** (continued)

13. Return the last character in the string:

We can return the last character in the string as the following:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    string st;
    cout << "Enter your string: ";
    cin >> st;
    cout << st.back() << endl;
    system("pause");
}
```



Library Functions for working with **strings** (continued)

13. Convert c-string into numeric data type:

A. C-string into integer value:

```
#include<iostream>
using namespace std;
void main()
{
    char ch[] = "1000";
    int num = atoi(ch);
    cout << num << endl;
    system("pause");
}
```




Library Functions for working with **strings** (continued)

14. Convert c-string into numeric data type:

B. C-string into long integer value:

```
#include<iostream>
using namespace std;
void main()
{
    char ch[] = "500000";
    long num = atol(ch);
    cout << num << endl;
    system("pause");
}
```



Library Functions for working with **strings** (continued)

14. Convert c-string into numeric data type:

C. C-string into float value:

```
#include<iostream>
using namespace std;
void main()
{
    char ch[] = "3.4";
    float num = atof(ch);
    cout << num << endl;
    system("pause");
}
```



Library Functions for working with **strings** (continued)

14. Convert c-string into numeric data type:

D. C-string into double value:

```
#include<iostream>
using namespace std;
void main()
{
    char ch[] = "12.67";
    double num = atof(ch);
    cout << num << endl;
    system("pause");
}
```



Library Functions for working with **strings** (continued)

15. Convert numeric value into string class:

```
#include<iostream>
#include<string>
using namespace std;
void main()
{
    int num = 1000;
    string st = to_string(num);
    cout << st << endl;
    system("pause");
}
```

How we can check if st is not empty string?

How we can convert numeric value into c-string?



Thank You ...