

Assignment # 4

Date: / /20

Day: Mon Tue Wed Thu Fri Sat

Name: Afraz Haider

SAP ID: 72935

Q. Why string functions are used in C++ and how
<cstring> supports low level string manipulation.

String functions are fundamentally used in C++ for the necessary manipulation of text data. The C-style string functions, defined in the <cstring> or <string.h> header files remain useful for several reasons. These functions operate directly on null-terminated character arrays, which is fundamental to how strings are represented at a low-level in C++.

The <cstring> library provides direct, low-level access and control over memory buffers containing strings. This is vital when interfacing with C libraries, operating systems, or code, which often require or return null-terminated C strings. Functions like `strlen()`, `strcpy()`, and `strcat()` allow for byte-by-byte manipulation and size management.

Teachers Signature: _____

substr() :- This function works with the

safe, dynamically allocated C++ string object.

It returns an entirely new string object containing
the requested segment based on a starting position

and length (optional). The original string
remains unchanged.

Code :- #include <iostream>

#include <string>

using namespace std;

int main() {

string intro = "My. Name.. is Afraz.";

string sub = intro.substr(3, 5);

string end = intro.substr(8);

cout << "Original : " << intro << endl;

cout << "Result 1: " << sub << endl;

cout << "Result 2: " << end << endl;

return 0; }

Output:

Original: My. Name.. is Afraz

Result 1: Name.

Result 2: is Afraz.

Teacher's Signature: _____

Date: / /20

Day: Mon Tue Wed Thu Fri Sat

strcat() :- strcat() appends the entire :source string to the end of the destination string. It overwrites the destination null terminator (\0) and appends the source string, followed by a new null terminator. The destination must be large enough to store the combined string.

Code: #include <iostream>

```
#include <cstring>
using namespace std;
int main() {
    char dest[30] = "Artificial";
    char source[15] = "Intelligence";
    cout << "Destination before: " << dest << endl;
    strcat(dest, source);
    cout << "Destination after: " << dest << endl;
    return 0;
}
```

output:

Destination before: Artificial

Destination after: ~~Intelligence~~ Artificial Intelligence

Teachers Signature: _____

Date: / /20

Day: Mon Tue Wed Thu Fri Sat

strcmp(): strcmp() compares two strings character by character based on their ASCII values until a difference is found or the end of both strings is reached. It returns 0 if identical, a negative number if the first string comes before the second; And a positive number if the first string comes after the second.

Code: #include <iostream>

#include <cstring>

using namespace std;

int main(){

char s1[5] = "Data";

char s2[5] = "Data";

char s3[5] = "Base";

int r1 = strcmp(s1, s2);

int r2 = strcmp(s2, s3);

int r3 = strcmp(s3, s1);

cout << "Data vs Data: " << r1 << endl;

cout << "Data vs Base: " << r2 << endl;

cout << "Base vs Data: " << r3 << endl;

return 0;

}

Teacher Signature:

Date: / /20

Day: Mon Tue Wed Thu Fri Sat

Output:-

Data vs Data : 0

Data vs Base : 1

Base vs Data : -1

Strcpy() :- strcpy() copies the source string, including its terminating null character (\0), byte-for-byte, into destination array, overwriting the destination content. The destination array must be large enough to store the copied content.

Code: #include <iostream>

#include <cstring>

using namespace std;

int main() {

char src[25] = "Low-Level Language";

char dest[30];

strcpy(dest, src);

cout << "Source: " << src << endl;

cout << "Destination: " << dest << endl;

return 0;

}

Output:

Source: Low-Level Language.

Destination: Low-level Language

strlen() :- strlen() calculates the length of a C-string by counting the number of characters from the start until it encounters the first null terminator. (\0).

The function's result does not include the null terminator itself.

Code: #include <iostream>

#include <cstring>

using namespace std;

int main() {

char str1[15] = "Programming";

char str2[2] = "a";

int len1 = strlen(str1);

int len2 = strlen(str2);

cout << "\n" << str1 << "\n Length : " << len1 << endl;

cout << "\n" << str2 << "\n Length : " << len2 << endl;

return 0;

{}

Date: / /20

Day: Mon Tue Wed Thu Fri Sat

Output:

"Programming" length: 11

"a" length: 1

strncpy(): This is a safer version of strcat(). It copies a maximum of N characters from the source string to the destination and always ensures the result is properly null-terminated. It limits the amount of data copied to prevent buffer overflow.

Code:

```
#include <iostream>
```

```
#include <cstring>
```

```
using namespace std;
```

```
int main() {
```

```
    char dest[20] = "Welcome";
```

```
    char source[20] = "to the C++ course";
```

~~strncpy(dest, source, 7);~~

```
cout << "Destination before: " << dest << endl;
```

```
strncpy(dest, source, 7);
```

```
cout << "Destination after: " << dest << endl;
```

```
return 0;
```

}

Teachers Signature: _____

Date: / /20

Day: Mon Tue Wed Thu Fri Sat

Output:

Destination before: Welcome

Destination after: Welcome to the

Code to Run:-

```
#include <iostream>
```

```
#include <ctype>
```

```
using namespace std;
```

```
int main()
```

```
{ char messystring[] = "t6Ho19S6.iS.999a9.STRING";
```

```
char current = messystring[0];
```

```
for (int i = 0; current != '\0'; current = messystring[++i]) {
```

```
if (isalpha(current))
```

```
cout << (char)(isupper(current) ? tolower(current) : current);
```

```
else if (ispunct(current))
```

```
cout << " ";
```

```
}
```

```
cout << endl;
```

```
return 0;
```

```
}
```

Teachers Signature: _____

Teachers Signature: _____

assignment 4 substr.cpp

```
1 #include<iostream>
2 #include<cstring>
3 using namespace std;
4 int main(){
5     string intro="My Name is Afraz";
6     string sub=intro.substr(3,5);
7     string end=intro.substr(8);
8
9     cout<<"Original: "<<intro<<endl;
10    cout<<"Result 1: "<<sub<<endl;
11    cout<<"Result 2: "<<end<<endl;
12    return 0;
13 }
```

C:\Users\Probook\Desktop\B! X + ▾

Original: My Name is Afraz

Result 1: Name

Result 2: is Afraz

Process exited after 0.5734 seconds with return value 0
Press any key to continue . . .

assignment 4 stringcat.cpp

X

[*] assignment 4

```
1 #include<iostream>
2 #include<cstring>
3 using namespace std;
4 int main(){
5     char dest[20] = "File ";
6     char src[8] = "System!";
7     cout << "Destination before:" << dest << endl;
8     strcat(dest, src);
9     cout << "Destination after:" << dest << endl;
10    return 0;
11 }
```

C:\Users\Probook\Desktop\B! X +

Destination before:File
Destination after:File System!

Process exited after 0.4876 seconds with return value 0
Press any key to continue . . .

assignment 4 strcmp.cpp

X

assignnn

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int main(){
5     char s1[5] = "Data";
6     char s2[5] = "Data";
7     char s3[5] = "Base";
8
9     int r1 = strcmp(s1, s2);
10    int r2 = strcmp(s1, s3);
11    int r3 = strcmp(s3, s1);
12
13    cout << "Data vs Data : " << r1 << endl;
14    cout << "Data vs Base : " << r2 << endl;
15    cout << "Base vs Data : " << r3 << endl;
16    return 0;
17 }
```



C:\Users\Probook\Desktop\B! X



Data vs Data : 0

Data vs Base : 1

Base vs Data : -1

Process exited after 0.4713 seconds with return value 0
Press any key to continue . . .

assignment 4 stringcpy.cpp

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int main() {
5     char src[25] = "Low-level Language";
6     char dest[30]; // Must be large enough
7
8     strcpy(dest, src);
9
10    cout << "Source: " << src << endl;
11    cout << "Destination: " << dest << endl;
12    return 0;
13 }
```

C:\Users\Probook\Desktop\B! X + | v

Source: Low-level Language

Destination: Low-level Language

Process exited after 0.6171 seconds with return value 0
Press any key to continue . . .

assignment 4 strlen.cpp

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int main() {
5     char str1[15] = "Programming";
6     char str2[2] = "a";
7
8     int len1 = strlen(str1); // 11 characters
9     int len2 = strlen(str2); // 1 character
10
11    cout << "\n" << str1 << "\n length: " << len1 << endl;
12    cout << "\n" << str2 << "\n length: " << len2 << endl;
13
14}
```

assignment 4 strncat.cpp

C:\Users\Probook\Desktop\B! X + ▾

```
"Programming" length: 11
"a" length: 1
```

```
Process exited after 0.4631 seconds with return value 0
Press any key to continue . . .
```

assignment 4 strncat.cpp

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 int main() {
5     char dest[20] = "Welcome";
6     char src[20] = " to the C++ course!";
7
8     cout << "Destination before: " << dest << endl;
9
10    strncat(dest, src, 7); |
11
12    cout << "Destination after: " << dest << endl;
13    return 0;
14 }
```

C:\Users\Probook\Desktop\B! ×



Destination before: Welcome

Destination after: Welcome to the

Process exited after 0.4383 seconds with return value 0
Press any key to continue . . .

rewritten code.cpp

x

```
1 #include <iostream>
2 #include <cctype>
3 using namespace std;
4 int main() {
5     char messyString[] = "t6H019s6.is.999a9.STRING";
6
7     char current = messyString[0];
8     for (int i = 0; current != '\0'; current = messyString [++i]) {
9         if (isalpha (current))
10             cout << (char) (isupper (current) ? tolower (current): current);
11         else if(ispunct (current))
12             cout << ' ';
13     }
14     cout<<endl;
15
16     cout<<"Afraz Haider"<<endl;
17     cout<<"72935"<<endl;
18     return 0;
19 }
```

C:\Users\Probook\Desktop\B! X + V

this is a string
Afraz Haider
72935

Process exited after 0.4412 seconds with return value 0
Press any key to continue . . .