Lecture Nine - String in C++

Introduction

C++ strings are sequences of characters stored in a char array. Strings are used to store words and text. They are also used to store data, such as numbers and other types of information. Strings in C++ can be defined either using the std::string class or the C-style character arrays.

C Style Strings

In C programming, the collection of characters is stored in the form of arrays terminated by a null character '\0'. This is also supported in C++ programming. Hence, it's called C-strings. They are the type of strings that C++ inherited from C language.

Syntax:

```
char str[] = "C++";
char str[4] = "C++";
char str[] = {'C','+','+','\0'};
char str[4] = {'C','+','+','\0'};
```

```
// C++ Program to demonstrate strings
#include <iostream>
using namespace std;
int main(){
    char s[] = "This is my program”;
    cout << s << endl;
    return 0;}
```

Output:

```
This is my program
```

Syntax:

```
std::string str("Good");
```

```
// C++ program to create std::string objects
#include <iostream>
using namespace std;
int main(){
    string str="Good Morning”;
```

```
    cout << str;
    return 0;}
```

Output:

```
Good Morning
```

How to Take String Input in C++

String input means accepting a string from a user. In C++. We have different types of taking input from the user which depend on the string. The most common way is to take input with cin keyword with the extraction operator (>>) in C++. Methods to take a string as input are:

1. Using Cin

The simplest way to take string input is to use the cin command along with the stream extraction operator (>>).

Syntax:

```
cin>>s;
```

Example:

```
// C++ Program to demonstrate string input using cin
#include <iostream>
using namespace std;
int main() {
      string s;
    cout<<"Enter String"<<endl;
      cin>>s;
      cout<<"String is: "<<s<<endl;
    return 0;}
```

Output:

```
Enter String
Hello!!!!
String is: Hello!!!!
```

C++ Program:

```
//C++ program to display a string entered by user.
#include <iostream>
using namespace std;
int main(){
    char str[100];
    cout << "Enter a string: ";
    cin >> str;
```

```
    cout << "You entered: " << str << endl;
    cout << "\nEnter another string: ";
    cin >> str;
    cout << "You entered: " << str << endl;
    return 0;}
```

Output 1:

```
Enter a string: C++
You entered: C++
Enter another string: Programming is fun.
You entered: Programming
```

Example 2:

To read the text containing blank space, cin.get function can be used. This function takes two arguments. The first argument is the name of the string (address of the first element of the string), and the second argument is the maximum size of the array.

```
#include <iostream>
using namespace std;
int main(){
    char str[100];
    cout << "Enter a string: ";
    cin.get(str, 100);
    cout << "You entered: " << str << endl;
    return 0;}
```

Output

```
Enter a string: Programming is fun.
You entered: Programming is fun.
```

2. Using getline

The getline() function in C++ is used to read a string from an input stream. It is declared in the <string> header file.

Syntax:

```
getline(cin,s);
// C++ Program to demonstrate use of getline function
#include <iostream>
using namespace std;
int main(){

    string s;
```

```
    cout << "Enter String" << endl;
    getline(cin, s);
    cout << "String is: " << s << endl;
    return 0;}
```

Output:

```
Enter String
Test Prog.
String is: Test Prog.
```

```
//C++ program string using string data type
#include <iostream>
using namespace std;
int main(){
    // Declaring a string object
    string str;
    cout << "Enter a string: ";
    getline(cin, str);
    cout << "You entered: " << str << endl;
    return 0;}
```

Output

```
Enter a string: Programming is fun.
You entered: Programming is fun.
In this program, a string str is declared. Then, the string
is asked from the user.
```

Instead of using cin>> or cin.get() function, you can get the entered line of text using getline(). getline() function takes the input stream as the first parameter, which is cin and str as the location of the line to be stored.

C++ String Functions

C++ provides some inbuilt functions which are used for string manipulation, such as the strcpy() and strcat() functions for copying and concatenating strings. Some of them are:

Function	Description
length()	This function returns the length of the string.
swap()	This function is used to swap the values of 2 strings.
size()	Used to find the size of string
resize()	This function is used to resize the length of the string up to the given number of characters.
find()	Used to find the string which is passed in parameters
push_back()	This function is used to push the passed character at the end of the string
pop_back()	This function is used to pop the last character from the string
clear()	This function is used to remove all the elements of the string.
strncmp()	This function compares at most the first num bytes of both passed strings.
strncpy()	This function is similar to strcpy() function, except that at most n bytes of src are copied
strrchr()	This function locates the last occurrence of a character in the string.
strcat()	This function appends a copy of the source string to the end of the

Function	Description
	destination string
find()	This function is used to search for a certain substring inside a string and returns the position of the first character of the substring.
replace()	This function is used to replace each element in the range [first, last) that is equal to old value with new value.
substr()	This function is used to create a substring from a given string.
compare()	This function is used to compare two strings and returns the result in the form of an integer.
erase()	This function is used to remove a certain part of a string.

You need to add the statement #include <strings> to the beginning of any program that uses a str... function

Name	*Operation*
`int strlen(string)`	Returns the number of characters in a string.
`void strcpy(target, source, n)`	Copies the source string into a target array.
`void strcat(target, source, n)`	Concatenates the source string onto the end of the target string.
`void strncpy(target, source, n)`	Copies a string up to *n* characters from the source string into a target array.

`void strncat(target, source, n)`	Concatenates the source string onto the end of the target string or 'n' characters, whichever comes first.
`int strnstr(string, pattern, n)`	Finds the first occurrence of one pattern string in another.
`int strncmp(source1, source2, n)`	Compares the first *n* characters in two strings. Returns a zero if the two strings match exactly.
`int strnicmp(source1, source2)`	Compares up to *n* characters in two strings without regard to case.

C++ String strcmp Example

```
#include <iostream>
#include <cstring>
using namespace std;
int main () {
  char key[] = "mango";
  char buffer[50];
  do {
     cout<<"What is my favourite fruit? ";
     cin>>buffer;
  } while (strcmp (key,buffer) != 0);
 cout<<"Answer is correct!!"<<endl;
  return 0; }
```

Output:

```
What is my favourite fruit? apple
What is my favourite fruit? banana
What is my favourite fruit? mango
Answer is correct!!
```

C++ String Concat Example

Let's see the simple example of string concatenation using strcat() function.

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
```

```
    char key[25], buffer[25];
    cout << "Enter the key string: ";
    cin.getline(key, 25);
    cout << "Enter the buffer string: ";
     cin.getline(buffer, 25);
    strcat(key, buffer);
    cout << "Key = " << key << endl;
    cout << "Buffer = " << buffer<<endl;
    return 0; }
```

Output:

```
Enter the key string: Welcome to
Enter the buffer string: C++ Programming.
Key = Welcome to C++ Programming.
Buffer = C++ Programming.
```

C++ String Copy Example

Let's see the simple example of copy the string using strcpy() function.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()  {
    char key[25], buffer[25];
    cout << "Enter the key string: ";
    cin.getline(key, 25);
    strcpy(buffer, key);
    cout << "Key = "<< key << endl;
    cout << "Buffer = "<< buffer<<endl;
    return 0; }
```

Output:

```
Enter the key string: C++ Tutorial
Key = C++ Tutorial
Buffer = C++ Tutorial
```

C++ String Length Example

Let's see the simple example of finding the string length using strlen() function.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()  {
char ary[] = "Welcome to C++ Programming";
```

```
cout << "Length of String = " << strlen(ary)<<endl;
return 0; }
```

Output:

```
Length of String = 26
```

gets() and puts()

Functions gets() and puts() are two string functions to take string input from the user and display it respectively as mentioned in the previous chapter.

```
#include<iostream>
int main(){
char name[30];
cout<<"Enter name: ";
gets(name);      //Function to read string from user.
cout<<"Name: ";
puts(name);     //Function to display string.
return 0;}
```

Note: Though, gets() and puts() function handle strings, both these functions are defined in "string.h" header file.

The standard C++ library provides a string class type that supports all the operations mentioned above, additionally much more functionality. Let us check the following example:-

```
#include <iostream>
#include <string>
int main () {
   string str1 = "Hello";
   string str2 = "World";
   string str3;
   int  len ;
   str3 = str1; // copy str1 into str3
   cout << "str3 : " << str3 << endl;
   // concatenates str1 and str2
   str3 = str1 + str2;
   cout << "str1 + str2 : " << str3 << endl;
   // total length of str3 after concatenation
   len = str3.size();
   cout << "str3.size() :  " << len << endl;
return 0;}
```

When the above code is compiled and executed, it produces result something as follows –

```
str3 : Hello
str1 + str2 : HelloWorld
str3.size() :  10
```

Homework (String in C++)

1. Write C++ Program to find the frequency of a character in a string
2. Write C++ Program to find the number of vowels, consonants, digits and white spaces
3. Write C++ Program to reverse a string using recursion
4. Write C++ Program to find the length of a string
5. Write C++ Program to concatenate two strings
6. Write C++ Program to copy a String
7. Write C++ Program to remove all characters in a string except alphabets
8. Write C++ Program to sort elements in the lexicographical order (dictionary order)
9. Write a C++ program to display a string from backward.
10. Write a C++ program to count number of words in string.