

OBJECT: To Understand C-Strings, C-String Variables, Reading Embedded Blanks, Reading Multiple Lines, Arrays Of Strings, String Class In C++, Defining And Accessing String Objects And Understanding Some String Class Member Functions.

STRINGS

Strings are used to represent text. They are set of characters that can also contain alphabets, spaces, numbers and other symbols. C++ provides following two types of string representations:

- The C-style character string.
- The string class type introduced with Standard C++.

C-STRINGS

The C-String originated within the C language and continues to be supported within C++. This string is actually a one-dimensional array of characters which is terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null.

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

Declaration Syntax:

```
char str [Size];
```

Initialization:

```
char str [] = {'H','e','l','l','o','\0'};
```

OR

```
char str[] = "Hello";
```

Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

INPUT/OUTPUT WITH C-STRINGS

Input: `cin>> str ;` **Output:** `cout<<str ;`

To access single character: `cout<<str[4];`

Reading Embedded Blanks

```
cin.get(str, SIZE);
```

Here **str** represents string array and **SIZE** is maximum number of characters

Reading Multiple Lines

```
cin.get(str, SIZE, TERMINATING_CHARACTER);
```

Here **str** represents string array, **SIZE** is maximum number of characters and **TERMINATING_CHARACTER** is a single character which stops the input

PROGRAM 1: Demonstrate Simple C-string

```
#include <iostream>
using namespace std;
int main()
{
char str1[] = {'H','e','l','l','o','\0'};
char str2[] = "Welcome";

cout<<str1<<" "<<str2<<endl;
char str3[20]; //string variable str
cout << "Enter a string: ";
cin.get(str3, 20); //put string in str
cout << "You entered: " << str3 << endl;
return 0;
}
```

PROGRAM 2: Demonstrate multiline input

```
#include <iostream>
using namespace std;
const int MAX = 2000;
char str[MAX];
int main()
{
cout << "Enter a string:"<<endl;
cin.get(str, MAX, '$'); //terminate with $
cout << "You entered:"<<endl << str << endl;
return 0;
}
```

PROGRAM 3: Demonstrate array of String

```
#include <iostream>
using namespace std;
int main()
{
const int DAYS = 7;
const int MAX = 10;
char star[DAYS][MAX] = { "Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday" };
for(int j=0; j<DAYS; j++)
cout << star[j] << endl;
return 0;
}
```

THE STRING CLASS IN C++

The standard C++ library provides a string class type that supports all the operations mentioned above, additionally much more functionality.

DEFINING AND ASSIGNING STRING OBJECTS

You can define a string object in several ways:

```
string s1("Man");  
string s2 = "Beast";  
string s3;
```

You can concatenate one string object with another:

```
s3 = "Neither " + s1 + " nor ";
```

You can also use the += operator to append a string to the end of an existing string:

```
s3 += s2;
```

You can exchange the values of strings:

```
s1.swap(s2);
```

INPUT/OUTPUT WITH STRING OBJECTS

Input: `cin>> s1 ;` Output: `cout<<s1 ;`

To access single character: `cout<<str[4];`

Reading Embedded Blanks

```
getline(cin, str);
```

First argument is stream object from which the input will come in this case (**cin**) and second argument **str** represents string object

Reading Multiple Lines

```
getline(cin, str, TERMINATING_CHARACTER);
```

First argument is stream object from which the input will come in this case (**cin**), second argument **str** represents string object and **TERMINATING_CHARACTER** is a single character which stops the input

PROGRAM 4: Demonstrate defining and assigning string objects

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s1("Man"); //initialize
    string s2 = "Beast"; //initialize
    string s3;
    s3 = s1; //assign
    cout << "s3 = " << s3 << endl;
    s3 = "Neither " + s1 + " nor "; //concatenate
    s3 += s2; //concatenate
    cout << "s3 = " << s3 << endl;
    s1.swap(s2); //swap s1 and s2
    cout << s1 << " nor " << s2 << endl;
    return 0;
}
```

PROGRAM 5: Demonstrate string class input/output

```
#include <iostream>
#include <string> //for string class
using namespace std;
int main()
{ //objects of string class
    string full_name, nickname, address;
    string greeting("Hello, ");
    cout << "Enter your full name: ";
    getline(cin, full_name); //reads embedded blanks
    cout << "Your full name is: " << full_name << endl;
    cout << "Enter your nickname: ";
    cin >> nickname; //input to string object
    greeting += nickname; //append name to greeting
    cout << greeting << endl; //output: "Hello, Jim"
    cout << "Enter your address on separate lines\n";
    cout << "Terminate with '$'\n";
    getline(cin, address, '$'); //reads multiple lines
    cout << "Your address is: " << address << endl;
    return 0;
}
```

SOME STRING FUNCTIONS

- The `length()` member function, which returns the number of characters in the string object.
- The `find()` function looks for the string used as its argument in the string for which it was called.
- The `find_first_of()` function looks for any of a group of characters, and returns the position of the first one it finds.
- The function `find_first_not_of()` finds the first character in its string that is not one of a specified group.
- The `erase()` function removes a substring from a string. Its first argument is the position of the first character in the substring, and the second is the length of the substring.
- The `replace()` function replaces part of the string with another string. The first argument is the position where the replacement should begin, the second is the number of characters in the original string to be replaced, and the third is the replacement string.
- The `insert()` function inserts the string specified by its second argument at the location specified by its first argument.
- The `append()` function add characters at the end of the sentence the first argument is the number of characters to append, and the second is the character to be appended.
- The `substr()` member function. It returns a substring of the string for which it was called. Its first argument is the position of the substring, and the second is the number of characters. If second argument is not passed, string till end is taken as substring
- The `clear()` function deletes all character from string

PROGRAM 6: Demonstrates finding substrings in string objects

```
#include <iostream>
#include <string>
using namespace std;
int main(){
    string s1 = "We all should respect everyone";
    cout<<s1<<endl;
    cout<<"Total Characters: "<<s1.length()<<endl;
    int n;
    n = s1.find("respect");
    cout << "Found respect at index: " << n << endl;
    n = s1.find_first_of("abcd");
    cout << "First of spde at index: " << n << endl;
    n = s1.find_first_not_of("aeiouAEIOU");
    cout << "First consonant at: " << n << endl;
    return 0;
}
```

PROGRAM 7: Demonstrates changing parts of string objects

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string s1("Quick! call the Police.");
    string s2("Doctor");
    string s3("Don't ");
    s1.erase(0, 7); //remove "Quick! "
    s1.replace(9, 6, s2); //replace "Police" with "Doctor"
    s1.replace(0, 1, "C"); //replace 'S' with 's'
    s1.insert(0, s3); //insert "Don't " at beginning
    s1.erase(s1.size()-1, 1); //remove '.'
    s1.append(3, '!'); //append "!!!"
    int x = s1.find(' '); //find a space
    while( x < s1.size() ) //loop while spaces remain
    {
        s1.replace(x, 1, "/"); //replace with slash
        x = s1.find(' '); //find next space
    }
    cout << "s1: " << s1 << endl;
    return 0;
}
```

PROGRAM 8: Demonstrates substring of string objects

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string s1 = "His Name is Rizwan";
    string s2 = s1.substr(12);
    cout<<s1<<endl;
    cout<<s1.substr(4,4)<<": "<<s2;
    return 0;
}
```

PASSING ARRAYS TO FUNCTIONS

Arrays can be used as arguments to functions. In a function declaration, array arguments are represented by the data type and size of the array. For Example:

```
void display(char[3][3]);
```

You can also write declaration without size:

```
void display(char[][3]);
```

Function doesn't need the size of the first dimension as two dimensional array is an array of arrays. It doesn't need to know how many elements there are, but it does need to know how big each element is. Following is the declaration of one dimensional array:

```
void somefunc( int elem[] );
```

PROGRAM 1: Demonstrates a double dimensional array by creating a board for tic-tac-toe by using a function which is taking array as an argument.

```
#include<iostream>
#include<conio.h>
using namespace std;

void display(char[3][3]);

int main(){
char arr[3][3]={{' ',' ',' ',' '},
                {' ',' ',' ',' '},
                {' ',' ',' ',' '}};
for (int i=0 ; i<3 ; i++)
    for (int j=0 ; j<3 ; j++)
    {
        display(arr);
        cout<<"Please Enter X or O";
        cin>>arr[i][j];
        system("cls");
    }
}

void display(char a[3][3])
{
    cout<<"\n\n\n  ----- \n";
    for (int i=0 ; i<3 ; i++){
        for (int j=0 ; j<3 ; j++)
            cout<<" | "<<a[i][j];
    }
    cout<<" | \n  ----- \n\n";
}
```

EXERCISE 1:

Write a simple program that asks the user for a Car registration number and then verifies that by:

- Using a length function to make sure 6 characters were entered, use cout for displaying error if user enters less or more letters.
- Check the first three character to make sure they are alphabetic. You can use the isalpha(char) function for this.
- Check the last three character to make sure they are numeric. You can use the isdigit(char) function for this.
- Include "ctype.h" for both isalpha() and isdigit() functions

```
Enter car registration number: GF121
Sorry Enter again
Enter car registration number: B121N1
Sorry Enter again
Enter car registration number: BAN123
Registration Number Verified
```

EXERCISE 2:

Write a program that reads a commercial website URL from user; you should expect that the URL starts with 'www.' and ends with '.com'

Retrieve the name of the site and output it. For instance, if the user inputs www.yahoo.com, your program should output yahoo.

```
Enter URL: www.youtube.com
Name of Site: youtube
```