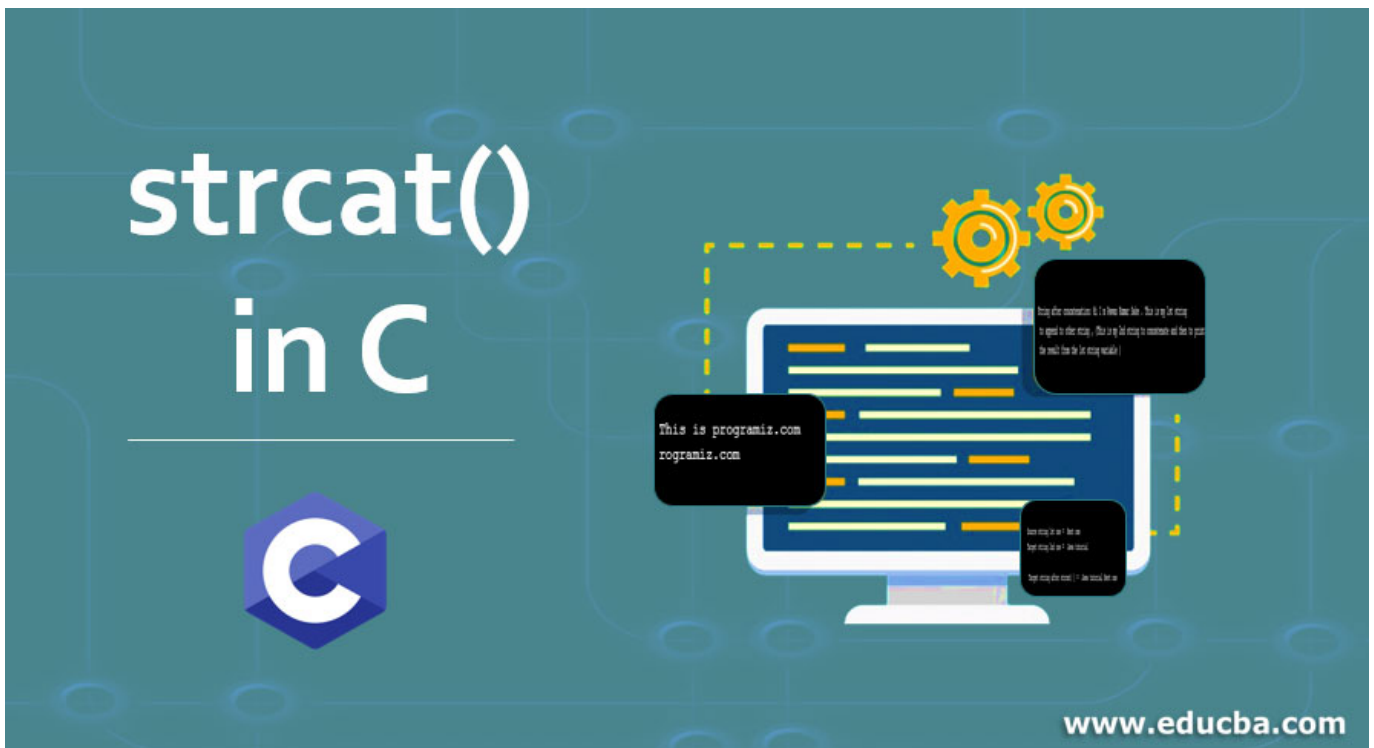




(<https://www.educba.com/software-development/>)

← (<https://www.educba.com/ascii-value-in-c/>)

→ (<https://www.educba.com/inline-function-in-c/>)



Introduction to strcat() in C

The `strcat()` function in C is usually used for the string concatenation in the programming process. `strcat()` concatenates a specific string with another string. It is a built-in function. `Strcat()` built-in function in C will only work without any error if you define `<string.h>` in the





(<https://www.educba.com/software-development/>)

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

```
Char *strcat(char *str1, const char *str2)
```

Parameters:

The above built-in function (strcat()) has/takes just 2 arguments which may be two strings/the character arrays. Strings that are concatenated will be stored in the first string itself in the argument.

- **Str1:** destination string's pointer.
- **Str2:** source string's pointer which can be appended/added to the destination string.

Examples to Implement strcat() in C

Below are the different examples of strcat() in C programming:

Example #1

Program to concatenate the strings. Here str1, str2 already assigned with the string values by assigning.

Code:

```
#include <stdio.h>
#include <string.h>
int main()
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
strcat(stra1,stra2);  
puts(stra1);  
puts(stra2);  
return 0;  
}
```

Output:

```
This is programiz.com  
rogramiz.com
```

Example #2

Program to concatenate stra1 and stra2 string's variable values.

Code:

```
#include <stdio.h>  
#include <string.h>  
int main () {  
    char stra1[1000], stra2[1000];  
    //stra1 destination string  
    strcpy(stra1, "Hi I m Pavan Kumar Sake . This is my 1st string to  
    append to other string , \n");  
  
    //stra2 source string  
    strcpy(stra2, " (This is my 2nd string to concatenate and then to  
    print the result from the 1st string variable )\n" );
```





(<https://www.educba.com/software-development/>)

```
return 0;
}
```



Popular Course in this category



C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,604 ratings)

Course Price

\$79 ~~\$399~~

[View Course](#)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

Output:

```
String after concatenation: Hi I m Pavan Kumar Sake . This is my 1st string to append to other
(This is my 2nd string to concatenate and then to print the result from the 1st string variable)
```



Example #3



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include <stdio.h>
#include <string.h>
int main () {
char str1[1000], str2[1000];
//str1 destination string
strcpy(str1, "Hi I m Pavan Kumar Sake . This is my 1st string \n
to append to other string , ");
//str2 source string
strcpy(str2, "(This is my 2nd string to concatenate and then to
print \n the result from the 1st string variable )" );
//concatenating the string str2 to the string str1 in the printf
statement itself
printf("String after concatenation: %s", strcat(str1, str2));
//The resultant of the concatenated string usually stores in str1
but here strcat() used in the print statement itself
//so no need to declare str1 again
return(0);
}
```

Output:

```
String after concatenation: Hi I m Pavan Kumar Sake . This is my 1st string
to append to other string , (This is my 2nd string to concatenate and then to print
the result from the 1st string variable )
```

Same output as the above example's output due to the same programming logic except declaration of the strcat() position. The same strings are used just like the above.



Example #4



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include <stdio.h>
#include <string.h>
int main () {
    //Assigning strings to the variables
    char str1[1000] = "Hello I m Pavan Kumar Sake . This is my 1st
    string to append to other string , \n";
    char str2[1000] = " (This is my 2nd string to concatenate and
    then to print the result from the 1st string variable )";
    //concatenating the string str2 to the string str1 in the printf
    statement itself
    printf("String after concatenation: %s", strcat(str1, str2));
    //The resultant of the concatenated string usually stores in str1
    but here strcat() used in the print statement itself
    //so no need to declare str1 again
    return(0);
}
```

Output:

```
String after concatenation: Hello I m Pavan Kumar Sake . This is my 1st string to append to other string ,
(This is my 2nd string to concatenate and then to print the result from the 1st string variable )
```

Example #5

C Program to show concatenation of 2 strings using 2 different types i.e., normal & also shifting the source and the target strings. Check the c program below, you will know.



Code



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
{
char source1[ ] = " Best one" ;
//assigning the string to the source1 string
char target1[ ]= " Java tutorial" ;
//assigning the string to the target1 string
printf ( "\nSource string 1st one = %s ", source1 ) ;
//printing the source1 value
printf ( "\nTarget string 2nd one = %s \n", target1 ) ;
//printing the target value
strcat ( target1, source1 ) ;
//concatenating the target1 and the source1 values/strings
printf ( "\n Target string after strcat( ) = %s \n ", target1 ) ;
//target1 will now have the concatenated and then it will be
printed
printf ( "\n Source string after strcat( ) = %s \n ",
strcat(source1, target1) ) ;
//source 1 doesnot have any value because already string appended
and the resultant stored in the target string/chars
}
```

Output:

```
Source string 1st one = Best one
Target string 2nd one = Java tutorial

Target string after strcat( ) = Java tutorial Best one
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include <stdio.h>
#include <string.h>
//Program to Concatenate the NULL STRING
#define DEST_SIZE1 40
// defining the dest_size1 of 40
char dest1[DEST_SIZE1];
//assigning the dest with dest size
int main()
{
    strcat(dest1, "Look Here");
    //concatenating the null string/character
    printf(dest1);
    //Printing the dest1 value
    return 0;
}
```

Output:

A black rectangular box with the text "Look Here" in a white, monospaced font, representing the output of the C program.

Look Here

Example #7

Program of concatenation using Pointers.

Code:

```
#include <stdio.h>
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
1
char src1[] = "Look Here";
//assigning string to the src1 variable
char dest1[DEST_SIZE1] = "Unimaginable";
//assigning string to the dest1 variable
char *ps1 = src1 + 4;
//assigning src1 string variables value to the pointer ps1
char *pd1 = dest1 + 6;
//assigning dest1 string variables value to the pointer pd1
strcat(pd1, ps1);
//concatenating the pointers
printf(dest1);
//printing dest1 value
strcat(src1, dest1);
//concatenating the src1 value by concatenating the src1 and dest1
printf("\n\n");
//line breaks
printf(src1);
//printing src1 value - it is the concatenation value
return 0;
}
```

Output:

```
Unimaginable Here
Look HereUnimaginable Here
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
#include <stdio.h>
#include <string.h>
//This is a program of strncat() function to concatenate only
first 4 characters of the string.
#define DEST_SIZE1 40
//defining the dest_size1 with 40 value
int main()
{
char src1[] = " Hey Buddy!!! ";
//assigning the string to the src character's string - 1st string
char dest1[DEST_SIZE1] = "How are you";
//assigning the string value to the dest[40] because dest_size1
already assigned - 2nd string
strncat(dest1, src1, 4);
//concatenating the dest, src with the condition of concatenating
only 4 strings of src1
printf(dest1);
//printing the dest1 value because now the dest1 now has
concatenated string
return 0;
}
```

Output:

```
How are you Hey
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

strcat are listed above.

Recommended Articles

This is a guide to strcat() in C. Here we discuss the syntax, Parameters and top examples of strcat() in C along with code implementation. You may also look at the following articles to learn more-

1. [Examples of C Union \(https://www.educba.com/c-union/\)](https://www.educba.com/c-union/)
2. [2-D Arrays in C \(https://www.educba.com/2-d-arrays-in-c/\)](https://www.educba.com/2-d-arrays-in-c/)
3. [Decimal to Octal in C \(https://www.educba.com/decimal-to-octal-in-c/\)](https://www.educba.com/decimal-to-octal-in-c/)
4. [Function Prototype in C \(https://www.educba.com/function-prototype-in-c/\)](https://www.educba.com/function-prototype-in-c/)

C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

- ☒ 3 Online Courses
- ☒ 5 Hands-on Projects
- ☒ 34+ Hours
- ☒ Verifiable Certificate of Completion
- ☒ Lifetime Access

Learn More

<https://www.educba.com/software-development/courses/c-programming-course/?bt=blg-inline-banner3>





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

About Us

Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)

Certificate from Top Institutions (<https://www.educba.com/educbalive/?source=footer>)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

Certification Courses

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)

Software Development Course - All in One Bundle (<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)

Become a Python Developer (<https://www.educba.com/software-development/courses/python-certification-course/?source=footer>)

Java Course (<https://www.educba.com/software-development/courses/java-course/?source=footer>)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.



