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

← (https://www.educba.com/fputs-in-c/)

→ (https://www.educba.com/fprintf-in-c/)

Introduction to C puts() Function

In any programming language, trying (https://www.educba.com/best-programming-lan
the output on the screen is a vital part. In c this is achieved with the help of puts function. Puts is an inbuilt function that writes a line of output to the screen. It returns the number of

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

The syntax of the puts function is as follows

```
int puts(const char *str)
```

The parameter string is the text that should be printed on the console.

Example of puts

Below the given codes and outputs explain the example in puts:

Code:

```
#include <stdio.h>
#include <string.h>
void main () {
char op1[30];
char op2[45];
strcpy(op1, "this is first line of puts example");
strcpy(op2, "this is second line of puts example");

puts(op1);
puts(op2);
}
```

gets Function in C

The gets function is a built-in function that is used to read the characters from the console and store that in a string. It reads the characters till a new line is found or EOF is reached, whichever comes first.

Syntax:

```
char *gets(char *str)
```

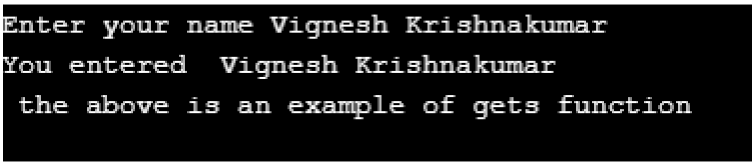
str is the pointer in which the character read is stored.

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

Code:

```
#include<stdio.h>
void main ()
{
char test[30];
printf("Enter your name");
gets(test);
printf("You entered %s",test);
printf("\n the above is an example of gets function");
}
```

Output:



```
Enter your name Vignesh Krishnakumar
You entered  Vignesh Krishnakumar
 the above is an example of gets function
```

Example of puts and gets

Below the given codes and outputs explain the example in puts and gets:

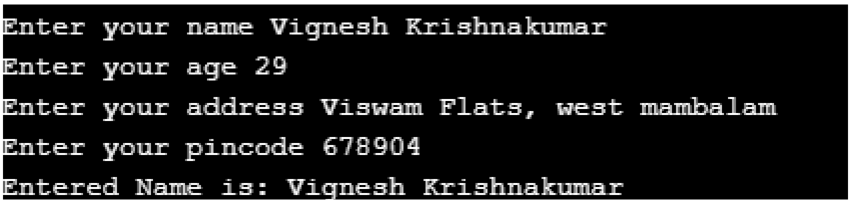
Code:

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

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

```
int age[50];
printf("Enter your age ");
gets(age);
char address[50];
printf("Enter your address ");
gets(address);
int pincode[50];
printf("Enter your pincode ");
gets(pincode);
printf("Entered Name is: ");
puts(name);
printf("Entered age is: ");
puts(age);
printf("Entered address is: ");
puts(address);
printf("Entered pincode is: ");
puts(pincode);
getch();
return 0;
}
```

Output:



```
Enter your name Vignesh Krishnakumar
Enter your age 29
Enter your address Viswam Flats, west mambalam
Enter your pincode 678904
Entered Name is: Vignesh Krishnakumar
```

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

The following points explain the function sin puts and gets with examples:

1. Fgetc()

This is an inbuilt file handling function (https://www.educba.com/file-handling-in-java/) that reads a file. It reads the file single character at a time until the end of the file. The function returns a non-negative value for successful execution.

Syntax:

```
Int fgetc(FILE *fp)
```

The fp is the pointer to the file location.

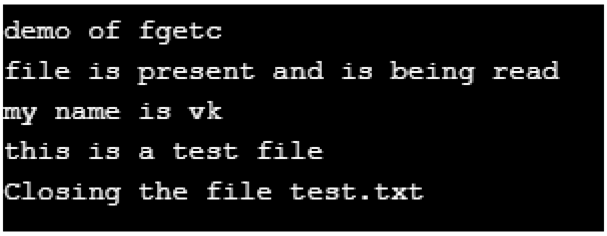
Code:

```
# include <stdio.h>
int main( )
{
FILE *testfile ;
char c ;
printf( "demo of fgetc" ) ;
testfile = fopen ( "c:\test.txt", "r" ) ;
if ( testfile == NULL )
{
printf ( "\nCould not open file test.txt" ) ;
return 1;
```

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

```
c = fgetc ( testfile ) ;
if ( c == EOF )
break ;
printf ( "%c", c ) ;
}
printf("Closing the file test.txt") ;
fclose ( testfile ) ;
return 0;
}
```

Output:



```
demo of fgetc
file is present and is being read
my name is vk
this is a test file
Closing the file test.txt
```

2. Fputc()

This is an inbuilt file handling function that writes (https://www.educba.com/file-handling-in-c-sharp/) text to a file. It writes one character at a time. The function returns a non-negative value for successful execution.

Syntax:

```
int fputc(int char, FILE *fp)
```

Where char is the character to be written and fp is the location of the file.

```
int main()
{
char ch;
FILE *source;
FILE *destination;
if (source = fopen("c:\source.txt", "r"))
{
ch = getc(source);
destination = fopen("c:\dest.txt", "w+")
while (ch != EOF)
{
fputc(ch, destination);
ch = getc(source);
}
fclose(source);
fclose(destination);
return 0;
}
return 1;
}
```

Output:

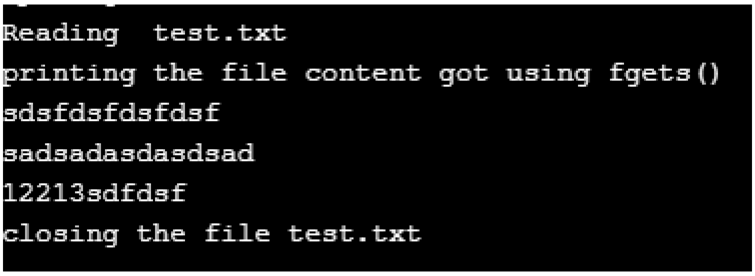
3. Fgets()

```
char *fgets(char *string, int n, FILE *fp)
```

- fp is the location from which the text must be read
- *string is the location in which the read characters are stored

Code:

```
# include <stdio.h>
int main( )
{
FILE *sourcefile ;
char newdata[50] ;
printf( "Opening  file test.txt " ) ;
sourcefile = fopen( "c:\test.c", "r" ) ;
if ( sourcefile== NULL )
{
printf( "Could not open file test.txt" ) ;
return 1;
}
printf( "Reading  test.txt" ) ;
while( fgets ( newdata, 50, sourcefile ) != NULL )
printf("\nprinting the file content got using fgets()")
printf( "%s" , newdata ) ;
printf("Closing the file test.txt") ;
fclose(sourcefile) ;
return 0;
```



```
Reading  test.txt
printing the file content got using fgets()
sdsfdsfdsfdsf
sadsadasdasdsad
12213sdfdsf
closing the file test.txt
```

4. Fputs()

This function is similar to fputc() except that this writes to a file line by line.

Syntax:

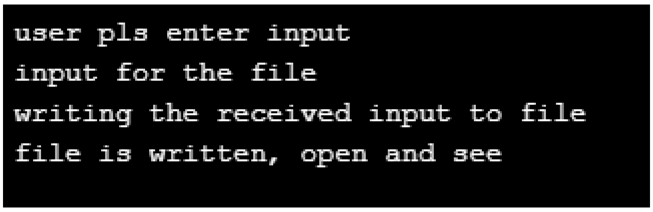
```
int fputs(const char *str, FILE *stream)
```

- *stream file to which the string must be written
- Str is the string that is to be written

Code:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
char input[50];
printf("\n user pls enter input");
scanf("%s",input);
FILE *destfile;
destfile = fopen("D:\test.txt", "w");
}
printf("\n writing the received input to file");
while( gets(input) != NULL )
{
fputs(input, destfile);
}
printf("\n file is written, open and see");
fclose(destfile);
return 0;
}
```

Output:



```
user pls enter input
input for the file
writing the received input to file
file is written, open and see
```

Difference Between

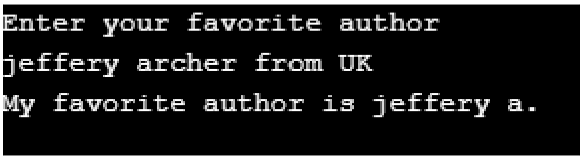
- **puts() and fputs():** Puts() is used to write the text to the console, whereas fputs() is used to write the text to a file. Puts() will convert a null character in the input to a new lin

  whereas fputs() will not handle the null character and stops execution.
- **gets() and fgets():** Gets reads the input from the console or stdin whereas fgets reads the input from a file. Fgets are safe to use as in case of overflowing it will throw an error.

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

```
#include<stdio.h>
int main()
{
char author[20];
printf("Enter your favorite author\n");
fgets(author,10,stdin);
printf("My favorite author is %s.\n",author);
FILE *destfile;
destfile = fopen("D:\existing.txt", "w+");
fputs("first line to be entered", destfile);
fputs("second line to be added", destfile);
fputs("thirdine to be added", destfile);
fclose(destfile);
return(0);
}
```

Output:



```
Enter your favorite author
jeffery archer from UK
My favorite author is jeffery a.
```

Conclusion – C puts() Function

Thus, the article covered in detail about puts functions along with its syntax and sample examples. It also covered in detail functions similar to it such as fputc() and fputs(). Also, functions like gets(), fgets() and fgetc() are discussed with appropriate examples. The

This is a guide to C puts() Function. Here we discuss detail about puts functions along with its syntax and sample examples, and function about fputc() and fputs(). You can also go through our other related articles to learn more –

1. Arithmetic Operators in C (https://www.educba.com/arithmetic-operators-in-c/)
2. Single Inheritance in C++ (https://www.educba.com/single-inheritance-in-c-plus-plus/)
3. strcmp() in C++ (https://www.educba.com/strcmp-in-c-plus-plus/)
4. if else Statement in C++ (https://www.educba.com/if-else-statement-in-c-plus-plus/)

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

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)

Resources

Free Courses (https://www.educba.com/software-development/free-cours
source=footer)

QUIZ

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

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

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)