

C Output Functions

C programming language provides built-in functions to perform output operation. The output operations are used to display data on user screen (output screen) or printer or any file. The c programming language provides the following built-in output functions...

1. **printf()**
2. **putchar()**
3. **puts()**
4. **fprintf()**

printf() function

The printf() function is used to print string or data values or a combination of string and data values on the output screen (User screen). The printf() function is built-in function defined in a header file called "**stdio.h**". When we want to use printf() function in our program we need to include the respective header file (stdio.h) using the **#include** statement. The printf() function has the following syntax...

Syntax:

```
printf("message to be display!!!");
```

Example Program

```
#include<stdio.h>

#include<conio.h>

void main(){

    printf("Hello! Welcome to btechsmartclass!!!");

}
```

Output:



"C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe"

Hello world!

Process returned 0 (0x0) execution time : 0.063 s

Press any key to continue.

In the above example program, we used the printf() function to print a string on to the output screen.

The printf() function is also used to display data values. When we want to display data values we use **format string** of the data value to be displayed.

Syntax:

```
printf("format string",variableName);
```

Example Program

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

```
#include<conio.h>
```

```
void main(){
```

```
    int i = 10;
```

```
    float x = 5.5;
```

```
    printf("%d %f",i, x);
```

```
}
```

Output:



"C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe"

```
10 5.500000
```

```
Process returned 0 (0x0)   execution time : 0.047 s
```

```
Press any key to continue.
```

In the above example program, we used the `printf()` function to print data values of variables `i` and `x` on to the output screen. Here `i` is an integer variable so we have used format string `%d` and `x` is a float variable so we have used format string `%f`.

The `printf()` function can also be used to display string along with data values.

Syntax:

```
printf("String format string",variableName);
```

Example Program

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

```
#include<conio.h>
```

```
void main(){
```

```
    int i = 10;
```

```
    float x = 5.5;
```

```
    printf("Integer value = %d, float value = %f",i, x);
```

```
}
```

Output:



"C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe"

```
Integer value = 10, float value = 5.500000
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

In the above program, we are displaying string along with data values.

Every function in the C programming language must have a return value. The printf() function also have an integer as a return value. The printf() function returns an integer value equivalent to the total number of characters it has printed.

Example Program

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

```
#include<conio.h>
```

```
void main(){
```

```
    int i;
```

```
    i = printf("btechsmartclass");
```

```
    printf(" is %d number of characters.",i);
```

```
}
```

Output:



"C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe"

```
btechsmartclass is 15 number of characters.  
Process returned 0 (0x0)   execution time : 0.031 s  
Press any key to continue.  
■
```

In the above program, first printf() function printing "btechsmartclass" which is of 15 characters. So it returns integer value 15 to the variable "i". The value of "i" is printed in the second printf() function.

Formatted printf() function

Generally, when we write multiple printf() statements the result is displayed in a single line because the printf() function displays the output in a single line. Consider the following example program...

Example Program

```
#include<stdio.h>

#include<conio.h>

void main(){

    printf("Welcome to ");

    printf("btechsmartclass ");

    printf("the perfect website for learning");

}
```

Output:



"C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe"

```
Welcome to btechsmartclass the perfect website for learning
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

In the above program, there are 3 printf() statements written in different lines but the output is displayed in single line only.

To display the output in different lines or as we wish, we use some special characters called **escape sequences**. Escape sequences are special characters with special functionality used in printf() function to format the output according to the user requirement. In the C programming language, we have the following escape sequences...

Escape sequence Meaning

\n	Moves the cursor to New Line
\t	Inserts Horizontal Tab (5 characters space)
\v	Inserts Vertical Tab (5 lines space)
\a	Beep sound
\b	Backspace (removes the previous character from its current position)
\\	Inserts Backward slash symbol
\?	Inserts Question mark symbol
\'	Inserts Single quotation mark symbol
\"	Inserts Double quotation mark symbol

Consider the following example program...

Example Program

```
#include<stdio.h>

#include<conio.h>

void main(){

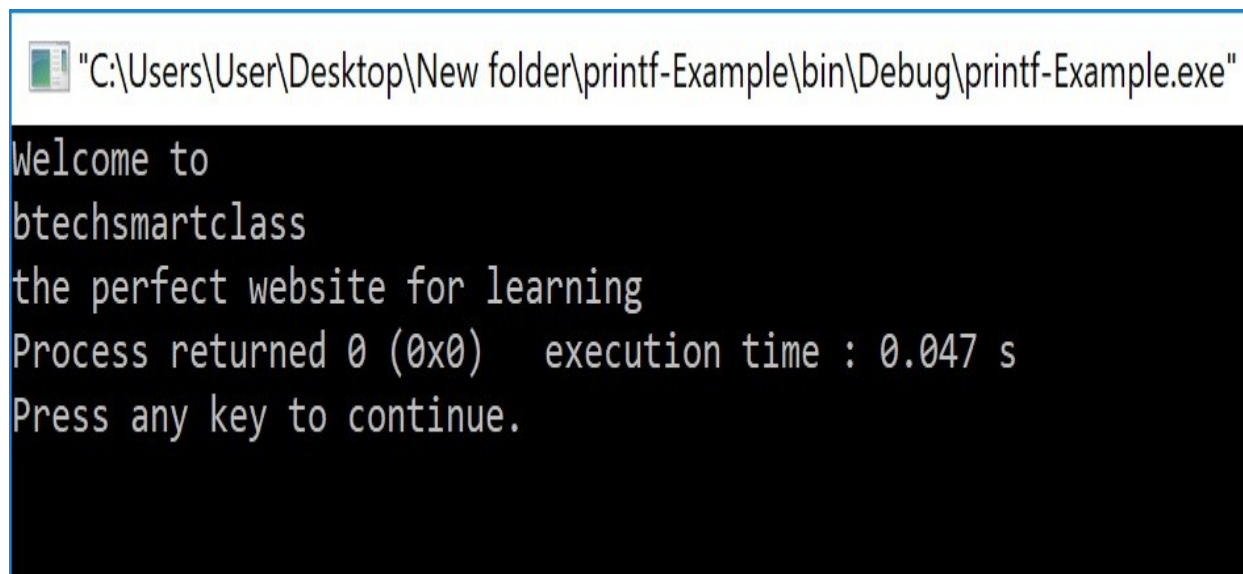
    printf("Welcome to\n");

    printf("btechsmartclass\n");

    printf("the perfect website for learning");

}
```

Output:

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe". The command prompt area has a black background with white text. The output of the program is displayed as follows: "Welcome to", "btechsmartclass", "the perfect website for learning", "Process returned 0 (0x0) execution time : 0.047 s", and "Press any key to continue.".

```
"C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe"

Welcome to
btechsmartclass
the perfect website for learning
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

putchar() function

The putchar() function is used to display a single character on the output screen. The putchar() functions prints the character which is passed as a parameter to it and returns the same character as a return value. This function is used to print only a single character. To print multiple characters we need to write multiple times or use a looping statement. Consider the following example program...

Example Program

```
#include<stdio.h>

#include<conio.h>

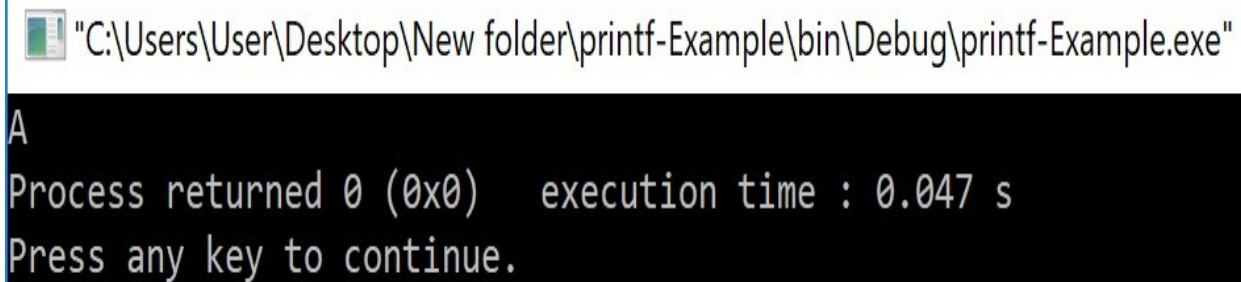
void main(){

    char ch = 'A';

    putchar(ch);

}
```

Output:



The screenshot shows a Windows command prompt window with a title bar that reads "C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe". The command prompt displays the output of the program: the character 'A' on the first line, followed by "Process returned 0 (0x0) execution time : 0.047 s" on the second line, and "Press any key to continue." on the third line.

puts() function

The puts() function is used to display a string on the output screen. The puts() functions prints a string or sequence of characters till the newline. Consider the following example program...

Example Program

```
#include<stdio.h>

#include<conio.h>

void main(){

    char name[30];

    printf("\nEnter your favourite website: ");

    gets(name);

    puts(name);

}
```

Output:



"C:\Users\User\Desktop\New folder\printf-Example\bin\Debug\printf-Example.exe"

```
Enter your favourite website: btechsmartclass
btechsmartclass
```

```
Process returned 0 (0x0)   execution time : 11.357 s
Press any key to continue.
```

fprintf() function

The fprintf() function is used with the concept of files. The fprintf() function is used to print a line into the file. When you want to use fprintf() function the file must be opened in writing mode.

