



C – PRINTF AND SCANF FUNCTIONS

Dr. Sheak Rashed Haider Noori
Professor & Associate Head
Department of Computer Science

C - PRINTF() AND SCANF() FUNCTIONS

- ❖ printf() and scanf() functions are inbuilt library functions in C which are available in C library by default.
- ❖ These functions are declared and related macros are defined in “stdio.h” which is a header file.
- ❖ We have to include “stdio.h” file as shown in below C program to make use of these printf() and scanf() library functions.

```
#include <stdio.h>

int main()
{
    int a, b, c;
    a = 5;
    b = 7;
    c = a + b;
    printf("%d + %d = %d\n", a, b, c);
    return 0;
}
```



```
#include <stdio.h>

int main()
{
    int a, b, c;
    a = 5;
    b = 7;
    c = a + b;
    printf("%d + %d = %d\n", a, b, c);
    return 0;
}
```

- ❖ Here is an explanation of the different lines in this program:
- ❖ The line **int a, b, c;** declares three integer variables named **a**, **b** and **c**. Integer variables hold whole numbers.
- ❖ The next line initializes the variable named **a** to the value 5.
- ❖ The next line sets **b** to 7.
- ❖ The next line adds **a** and **b** and "assigns" the result to **c**. The computer adds the value in **a** (5) to the value in **b** (7) to form the result 12, and then places that new value (12) into the variable **c**. The variable **c** is assigned the value 12. For this reason, the = in this line is called "the assignment operator."
- ❖ The **printf** statement then prints the line "5 + 7 = 12." The **%d** placeholders in the printf statement act as placeholders for values. There are three %d placeholders, and at the end of the printf line there are the three variable names: **a**, **b** and **c**. C matches up the first %d with a and substitutes 5 there. It matches the second %d with b and substitutes 7. It matches the third %d with c and substitutes 12. Then it prints the completed line to the screen: 5 + 7 = 12. The +, the = and the spacing are a part of the format line and get embedded automatically between the %d operators as specified by the programmer.

PRINTF() FUNCTION

❖ C printf() function:

- The **printf statement** allows you to send output to standard out. For us, standard out is generally the screen (although you can redirect standard out into a text file or another command).
- printf() function is used to print the “character, string, float, integer, octal and hexadecimal values” onto the output screen.
- We use printf() function with %d format specifier to display the value of an integer variable.
- Similarly %c is used to display character, %f for float variable, %s for string variable, %lf for double and %x for hexadecimal variable.
- To generate a newline, we use “\n” in C printf() statement.

❖ Note:

- C language is case sensitive. For example, printf() and scanf() are different from **Printf()** and **Scanf()**. All characters in printf() and scanf() functions must be in lower case.

EXAMPLE PROGRAM PRINTF() FUNCTION

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

```
int main()
```

```
{
```

```
    char ch = 'A';
```

```
    char str[20] = "fresh2refresh.com";
```

```
    float flt = 10.234;
```

```
    int no = 150;
```

```
    double dbl = 20.123456;
```

```
    printf("Character is %c \n", ch);
```

```
    printf("String is %s \n" , str);
```

```
    printf("Float value is %f \n", flt);
```

```
    printf("Integer value is %d\n" , no);
```

```
    printf("Double value is %lf \n", dbl);
```

```
    printf("Octal value is %o \n", no);
```

```
    printf("Hexadecimal value is %x \n", no);
```

```
    return 0;
```

```
}
```

Output:

Character is A

String is fresh2refresh.com

Float value is 10.234000

Integer value is 150

Double value is 20.123456

Octal value is 226

Hexadecimal value is 96



FORMAT SPECIFIER

❖ You can see the output with the same data which are placed within the double quotes of printf statement in the program except

- %d got replaced by value of an integer variable (no),
- %c got replaced by value of a character variable (ch),
- %f got replaced by value of a float variable (flt),
- %lf got replaced by value of a double variable (dbl),
- %s got replaced by value of a string variable (str),
- %o got replaced by a octal value corresponding to integer variable (no),
- %x got replaced by a hexadecimal value corresponding to integer variable
- \n got replaced by a newline.



EXAMPLE PROGRAM : PRINTF() AND SCANF() FUNCTIONS

❖ scanf() function:

- The **scanf function** allows you to accept input from standard in, which for us is generally the keyboard.
- scanf() function is used to read character, string, numeric data from keyboard
- Consider below example program where user enters a character. This value is assigned to the variable "ch" and then displayed.
- Then, user enters a string and this value is assigned to the variable "str" and then displayed.

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

```
int main()
```

```
{
```

```
    char ch;
```

```
    char str[100];
```

```
    printf("Enter any character \n");
```

```
    scanf("%c", &ch);
```

```
    printf("Entered character is %c \n", ch);
```

```
    printf("Enter any string ( upto 100 character ) \n");
```

```
    scanf("%s", &str);
```

```
    printf("Entered string is %s \n", str);
```

```
    return 0;
```

```
}
```

Enter any character

a

Entered character is a

Enter any string (upto 100 character)

hai

Entered string is hai



EXAMPLE PROGRAM : PRINTF() AND SCANF() FUNCTIONS

- ❖ The format specifier %d is used in scanf() statement. So that, the value entered is received as an integer and %s for string.
- ❖ Ampersand is used before variable name “ch” in scanf() statement as &ch , this operator is called *address of operator*.



BASIC IO FUNCTIONS

I/O Library Functions	Meanings
getch()	Inputs a single character (most recently typed) from standard input (usually console).
getche()	Inputs a single character from console and echoes (displays) it.
getchar()	Inputs a single character from console and echoes it, but requires <i>Enter</i> key to be typed after the character.
putchar() or putch()	Outputs a single character on console (screen).
scanf()	Enables input of formatted data from console (keyboard). Formatted input data means we can specify the data type expected as input. Format specifiers for different data types are given in Figure 21.6.
printf()	Enables obtaining an output in a form specified by programmer (formatted output). Format specifiers are given in Figure 21.6. Newline character "\n" is used in <i>printf()</i> to get the output split over separate lines.
gets()	Enables input of a string from keyboard. Spaces are accepted as part of the input string, and the input string is terminated when <i>Enter</i> key is hit. Note that although <i>scanf()</i> enables input of a string of characters, it does not accept multi-word strings (spaces in-between).
puts()	Enables output of a multi-word string

understanding the execution of a C Program