CSE115L – Programming Language I Lab

Lab-02

Data Types & Basic I/O

In this lab, we will learn about the basic input / output function in C. In C programming, the printf() function is one of the most commonly used output function. The function sends formatted output to the console. The following examples will help you remember the syntax.

General syntax of the printf() function:

printf("string literal");

String literal is a sequence of any number of characters surrounded by double quotation marks.

printf("format string", variables);

Format string is a combination of text, conversion specifier and escape sequence.

Escape sequences: Escape sequences are the special directives used to format printing. For example, \n indicates that the next printing should start from the first column of the next line. Escape sequences start with the backslash (\) character.

Escape Sequence	Effect
\a	Beep sound
\b	Backspace
\f	Formfeed (for printing)
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab
\\	Backslash
\"	" sign

Conversion specifier: A conversion specifier is a symbol that is used as a placeholder in a formatting string. For integer output (for example), %d is the specifier that holds the place for integers. Here are some commonly used conversion specifiers (not a comprehensive list):

Conversion Specifier	Output Type	Output Example
%d	Signed decimal integer	76
%u	Unsigned decimal integer	76
%o	Unsigned octal integer	134
%x	Unsigned hexadecimal (small letter)	9c
%X	Unsigned hexadecimal (capital letter)	9C
%f	Signed floating point (without e notation)	76.000000
%lf	Signed floating point (without e notation)	76.000000
%e	Signed floating point (using e notation)	7.6000e+01
%E	Signed floating point (using E notation)	7.6000E+01
%c	Character	'7'

Format specifiers: A format specifier can be used in conjunction with conversion specifiers to impose output formatting.

- A minus (-) sign tells left alignment.
- A number after % specifies the minimum field width to be printed if the characters are less than the size of width the remaining space is filled with space and if it is greater than it printed as it is without truncation. A zero before the number will fill the leading spaces with zeros.
- A period (.) symbol separates total field width from precision width.

Example 1: Using escape sequences in `printf()`

```
#include<stdio.h>

int main()
{
    printf("\t\"North South University\"\n");
    printf("Hello class of cse115L!!");
    printf("Welcome to NSU.\n");
    return 0;
}
```

Example 2: Using conversion specifiers in `printf()`

```
#include <stdio.h>

int main()
{
    int testInteger = 5;
    float testFloat = 123456.123456;
    double testDouble = 123456.123456;
    char testChar = 'a';

    printf("testInteger = %d\n", testInteger);
    printf("testFloat = %f\n", testFloat);
    printf("testDouble = %lf\n", testDouble);
    printf("testChar = %c", testChar);
    return 0;
}
```

Example 3: Using format specifiers in `printf()`

```
#include <stdio.h>

int main()
{
    printf("Preceding with blanks:\n%10d\n", 1977);
    printf("Preceding with zeros:\n%010d\n", 1977);
    printf("floats:\n%4.2f\n%+.0e\n%E\n", 3.1416, 3.1416, 3.1416);
    return 0;
}
```

In this lab, we will also learn about the basic input function in C. In C programming, the `scanf()` function is one of the most commonly used input function. The function is used to read characters, strings and numeric data from console. The following examples will help you remember the syntax.

General syntax of the `scanf()` function:

```
scanf("conversion specifier", &variable);
```

The following conversion specifiers should be used when using `scanf()`.

`%d` for integer input
`%f` for floating point number input
`%c` for character input
`%lf` for double number input

Example 4: Write a program that prompts the user to enter an integer value, a fractional number and a character and prints the entered value as output.

```
#include<stdio.h>
int main(){
    int num;
    float deci;
    char ch;

    printf("Enter a number:");
    scanf("%d",&num);
    printf("The number is %d\n",num);

    printf("Enter a fractional number:");
    scanf("%f",&deci);
    printf("The number is %.2f\n",deci);

    printf("Enter a character:");
    scanf("%c",&ch);
    printf("The character is: %c", ch);

    return 0;
}
```

Example 5: Write a program that reads in the radius of a circle and prints the circle's diameter, circumference and area.

```
#include<stdio.h>
int main(){
    float const PI = 3.142;
    float radius;
    float area, circumference, diameter;

    printf("Enter the radius of a circle:");
    scanf("%f",&radius);

    diameter= 2*radius;
    circumference= 2*PI*radius;
    area= PI * radius * radius;

    printf("The Diameter is: %.2f \n",diameter);
    printf("The Circumference is: %.2f \n",circumference);
    printf("The Area is: %.2f \n",area);
}
```

Perform the following tasks.

Task 1: Write a program where you declare an integer variable, a double variable and a character variable. Assign 139 to the integer, 5243.873649 to the double and 'X' to the character.

a) Print the integer variable as decimal, octal and hexadecimal values.
b) Print the double variable with E notation.
c) Print the double variable with 10 characters of total width and 3 characters of precision width. The leading spaces should be filled with zeros.
d) Print the character variable as both character and decimal.

Task 2: Ask user for two integers, a and b. Then swap (interchange) the values of a and b. That means, a should get the value of b and b should get the value of a.

Sample Input	Sample Output
Enter a: 3 Enter b: 7	After Swapping: a = 7, b = 3