C programming – Output statement

Output Statement [printf()] –

It is a library function which is used as an output function in C programming. It is an output function which displays the output on the standard output device.

1. printf("[Message]"):

   Using this function, we can print any string by putting it inside the double quotes (" ").

   For example: printf("We are learning C programming");

   OUTPUT: We are learning C programming

   printf("We are **\n**learning **\n**C programming");

   OUTPUT: We are

   learning

   C programming

   NOTE: (i) \n, \t, \b, \r are called Escape sequence.

   a. \n - New line/ Line break
   b. \t – Tab (Leaves 8 characters)
   c. \b – Backspace
   d. \r – Enter

   (ii) printf() returns the number of character inside the double quotes.

   For example: int x = printf("programming");

   So, x = 11

2. printf("[Conversion character]",[variable_name]):

   Using this function, we can print the value of any variable by using a suitable conversion character for the given variable.

   For Example: n=5;

   a. printf("%d",n);
      OUTPUT: 5
   b. printf("Value of n = %d",n);
      OUTPUT: Value of n = 5

Important conversion characters:

Variable type	Conversion character	Variable type	Conversion character
signed short int / short int	%hd, %hi	double	%lf, %le, %lg
unsigned short int	%u	String (Char array)	%s
signed int / int	%d, %i	char	%c
unsigned int	%u	float	%f, %e, %g
signed long int / long int / long	%ld,%li	unsigned long int	%u

NOTE: In float, %f %g and %e has different uses. For example:

N = 3.4753 can be expressed as:

%f – 3.4753 (Fixed point value)

%e – 34753E-4 (Exponential value)

%g – returns either of fixed point or exponential value

Points about conversion characters:

1. **For leaving spaces –**
   int n = 65;
   a. printf("n = %d", n);
      n = 65 (Normal form)
   b. printf("n = %10d", n);
      n = _ _ _ _ _ _ _ _ 65 (Using %10d, text appears after 10 places from backwards)
   c. printf("n = %-10d", n);
      n = 65_ _ _ _ _ _ _ _ (Using %-10d, text appears on the left of 10 spaces from starting)
2. **For after decimal values –**

   float n = 7.4678;

   a. printf("n = %f", n);
      n = 7.467800 (Normal form has 6 decimal values)
   b. printf("n = %.2f",n);
      n = 7.57 (Averages the decimal value upto 2 decimal places)
   c. printf("n = %10.2f",n);
      n = _ _ _ _ _ _ 7.57