ASSIGNMENT: OPERATOR

1) Bitwise Operator:

In arithmetic-logic unit (which is within the CPU), mathematical operations like: addition, subtraction, multiplication and division are done in bit-level. To perform bit-level operations in C programming, bitwise operators are used.

- a & b -- The AND operator compares two bits and generates a result of 1 if both bits are 1; otherwise, it returns 0.

- a | b -- The OR operator compares two bits and returns 1 if either of the bits are 1 and it gives 0 if both bits are 0 or 1.

- a^ b -- The EXCLUSIVE-OR operator compares two bits and generates a result of 1 if the bits are complementary; otherwise, it returns 0.

- ~a -- The COMPLEMENT operator is used to invert all of the bits of the operand.

- a >> b -- The SHIFT RIGHT operator moves the bits to the right, discards the far right bit, and assigns the leftmost bit a value of 0. Each move to the right effectively divides a in half.

- a << b -- The SHIFT LEFT operator moves the bits to the left, discards the far left bit, and assigns the rightmost bit a value of 0. Each move to the left effectively multiplies a by 2.

C Program to demonstrate use of bitwise operators

```
#include <stdio.h>
int main()
{
   unsigned char a = 5, b = 9;

   printf("a = %d, b = %d\n", a, b);
```

```
    printf("a&b = %d\n", a & b);

    printf("a|b = %d\n", a | b);
    printf("a^b = %d\n", a ^ b);
    printf("~a = %d\n", a = ~a);

    printf("b<<1 = %d\n", b << 1);
    printf("b>>1 = %d\n", b >> 1);
    return 0;
}
```

2) Ternary Operator:

If any operator is used on three operands or variable is known as Ternary Operator. It can be represented with **? :** . It is also called as conditional operator.

Syntax

```
expression-1 ? expression-2 : expression-3
```

3) PROGRAM TO PERFORM ARITHMETIC OPERATIONS:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a, b, sum, diff, product, div, mod;
    printf(" enter 2 numbers : ");
    scanf(" %d %d", &a, &b);
    sum = a+b;
```

```
    diff = a-b;
    product = a*b;
    div = a/b;
    mod = a%b;
    printf("sum of %d and %d is %d \n", a, b, sum);
    printf("diff of %d and %d is %d \n", a, b, diff);
    printf("product of %d and %d is %d \n", a, b, product);
    printf("div of %d and %d is %d \n", a, b, div);
    printf("mod of %d and %d is %d \n", a, b, mod);

}
```