**PRACTICAL # 03**

**OBJECT:**

Working with C# Operators

**THEORY:**

C# operators give a programmer detailed control over the construction and evaluation of expressions. Most of C#’s operators fall into these categories: arithmetic, bitwise, relational, and logical.

C# also defines operators that handle special situations, such as array indexing, member access, and the lambda operator.

**Program:**

This is arithmetic operators demo program in C#.

*//mod*

*int iresult, irem;*

*double dresult, drem;*

*iresult = 10 / 3;*

*irem = 10 % 3;*

*dresult = 10.0 / 3.0;*

*drem = 10.0 % 3.0;*

*//Increment and Decrement*

*int x=1, y=0;*

*int i;*

*Console.WriteLine("Series generated using y = y + x++;");*

*for(i = 0; i < 10; i++) {*

*y = y + x++; // postfix ++*

*Console.WriteLine(y + " ");*

*}*

*x = 1;*

*y = 0;*

*Console.WriteLine("Series generated using y = y + ++x;");*

*for(i = 0; i < 10; i++) {*

*y = y + ++x; // prefix ++*

*Console.WriteLine(y + " ");*

*}*

**ACTIVITIES**

**Activity 1**

Write a simple calculator program using the arithmetic operators.

**Activity 2**

Using C# bitwise operators, swap two int variables.

**REVIEW QUESTIONS**

1. What are different categories of operators in C#?
2. What is the difference between prefix and postfix operators?
3. How do bitwise operators work? Are they faster than normal operators?
4. Explain operator precedence with example.
5. What is the role of parenthesis in operator precedence?