Exercise 4
Operator Overloading

Objective

To develop C# Console applications using operator overloading concept.

Operator Overloading

Operator overloading gives the ability to use the same operator to do various operations. It provides additional capabilities to C# operators when they are applied to user-defined class types.

An operator can be overloaded by defining a function to it. The function of the operator is declared by using the operator keyword.

```
accessspecifier static returntype operator operator_symbol(parameters)
    {
        // Code
    }
```

Note: Operator overloading is basically the mechanism of providing a special meaning to an ideal C# operator with respect to a user-defined data classes.

Also, the operator overloading method must be a static method because an operator overload does not reference a class: it takes parameters and returns a value, but it does not have access to this object or any non-static members of the class.

Exercise Questions: You are expected to completed two Questions as given below

Question A: (Compulsory for All)
Create a **Matrix** class with a 2D Array as data to represent matrix values. Include constructor to initialize the **matrix (2x2)** using run time input and also include a method to display the matrix data representation. Create two Matrix objects such as M1, M2 and store the data using user input and perform the following matrix manipulation (Matrix Addition and Matrix Multiplication) using operator overloading and then display the result Matrix object data.

1. Generate a Matrix M3 = M1 + M2
2. Generate a Matrix M4 = M1 * M2

Question B: Select the question number using the formula, Q.No = Regno % 4 + 1

Q.No	Questions

1	Create a class called Complex to store complex numbers such as **real** and **imaginary** part. Initialize the data using constructor method and include a method to display the complex number in a format (for example 10 + i5). Create two objects as C1 and C2 with data initialization through run time input and demonstrate the operator overloading for the following operations and display the details of C3 object for every functionalities. • Calculate: C3 = C1 + C2 • Calculate: C3 = C1 * C2 • Calculate: C3 = C1 - C2 • Compare: C1 == C2
2	Create a class called **Time** with data hour, minutes, seconds and include the constructor function to initialize the data. Include another method to display the Time details in a proper format (Example: 03:30:45). Then, create two Time objects T1, T2 and initialize its data using user input and perform the following operations using operator overloading. • Calculate T3=T1+T2 • Calculate T3=T1-T2 • Check T1==T2 • Check T1>=T2
3	Create a class called **Currency** with two data such as Rupees and Paise. Include constructor method to initialize the data. Add a method to display the currency details. Create two Currency object C1, C2 and store the data using user input and perform the following operation using operator overloading. • Compute: C3 = C1 + C2 • Compute: C3 = C1 – C2 • Check: C1 > C2 • Check: C1 == C2
4	Design a class to represent a **Rectangle** with length and breadth as instance attributes. Create two rectangle objects, r1 and r2. Initialize the attributes using the constructor and do the following operations using operator overloading. • Compute R3 = R1 + R2, • Compare the dimensions of R1 and R2. ▪ R1 == R2 ▪ R1 < R2 ▪ R1 >= R2