**COMSATS University Islamabad, Attock Campus**



**Name: Ayesha Bibi**

**Reg No : FA20-BCS-044**

**Submitted to: Syed Bilal Bukhari**

**Midterm Lab**

**QUESTION NO: 01**

**Briefly describe the regex library of C#?**

**ANSWER:**

**DEFINATION:**

Regex in C# defines a regular expression in C#. Regex, short for regular expression, is a sequence of characters that defines a search pattern. The Regex class offers methods and properties to parse a large text to find patterns of characters.

In C#, the `Regex` class defines and handles regular expressions. Regular expressions are patterns of characters used for searching and manipulating text. They can consist of literal characters, metacharacters, character classes, quantifiers, and more.

**APPLICATIONS:**

Regular expressions are versatile tools with various applications:

1. **Pattern Matching:** They are used to find and extract specific character sequences within text.

2.  **String Validation:** Regular expressions help check if a string adheres to a particular pattern or format.

3. **Data Extraction:** They are useful for parsing and extracting data from structured text, like log files or web pages.

4. **Text Manipulation**: Regular expressions enable the replacement or modification of specific patterns within a string.

The `Regex` class in C# provides methods and properties for working with regular expressions, making it a valuable tool for text processing and manipulation tasks.

**LIBRARIES:**

This library enables you to work with and manipulate text patterns using regular expressions in C#. Within this library, you'll encounter essential components and classes:

1. **Regex Class**: This central class is the heart of regular expression operations in C#. It equips you with methods for pattern matching and text manipulation based on regular expressions.

2. **Match Class:** It represents an individual discovery within the input text. It allows access to groups and their respective captures.

3. **MatchCollection Class**: This class serves as a container for a collection of Match objects. It is often the result of methods like `Regex.Matches()`.

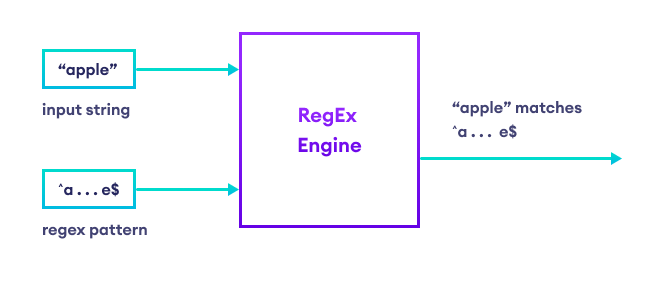
4. **Group Class:** This class delineates a captured group within a match. Groups are typically formed using parentheses in the regular expression pattern.

5. **Capture Class:** Each instance of Capture signifies a captured substring within a specific group.

Among the commonly used methods and properties of the `Regex` class are `Match()`, `Matches()`, `Replace()`, `Split()`, `IsMatch()`, and properties like `Options`, which allow you to specify various matching options.

**How Regex Works?**

In C#, there is an engine called regex engine which internally checks the regex pattern in the given string.



**EXAMPLE:**

Here's a simple illustration of the C# regex library in action:

using System;

using System.Text.RegularExpressions;

class Program

{

// a regular expression pattern for a five-letter word

// that starts with "a" and ends with "e"

static string pattern = "^a...e$";

static void Main()

{

// create an instance of Regex class and

// pass the regular expression (i.e pattern)

Regex rg = new Regex(pattern);

// IsMatch() returns true if "apple" matches the regular expression

if (rg.IsMatch("apple"))

{

Console.WriteLine("String matches the pattern");

}

else

{

Console.WriteLine("String doesn't match the pattern");

}

}

}

**Output**

String matches the pattern