C# (Some Important Terms)

INSTRUCTOR: ADEEN BIN WAHEED

LECTURE 8

Enumeration

An *enum type* is a special data type that enables for a variable to be a set of predefined constants. The variable must be equal to one of the values that have been predefined for it. Enumeration contains its own values and cannot inherit or cannot pass inheritance.

Common examples include compass directions (values of NORTH, SOUTH, EAST, and WEST) and the days of the week.

Because they are constants, the names of an enum type's fields are in uppercase letters.

In the C# programming language, you define an enum type by using the enum keyword. For example, you would specify a days-of-the-week enum type as:

```
enum Day {

   SUNDAY, MONDAY, TUESDAY, WEDNESDAY,

   THURSDAY, FRIDAY, SATURDAY

}
```

Enumeration

Running Demo

Structures

In C#, a structure is a value type data type. It helps you to make a single variable hold related data of various data types. The **struct** keyword is used for creating a structure.

Structures are used to represent a record. Suppose you want to keep track of your books in a library. You might want to track the following attributes about each book:

Title

Author

Subject

Book ID

Class VS Structure

Classes and Structures have the following basic differences:

- classes are reference types and structs are value types
- structures do not support inheritance
- structures cannot have default constructor

Structures

Running Demo

Strings in C#

In C#, you can use strings as array of characters, However, more common practice is to use the **string** keyword to declare a string variable. The string keyword is an alias for the **System.String** class.

Strings contains many functionalities.

public bool Contains(string value)

public bool Equals(string value)

public string Remove(int startIndex, int count)

public string Replace(char oldChar, char newChar)

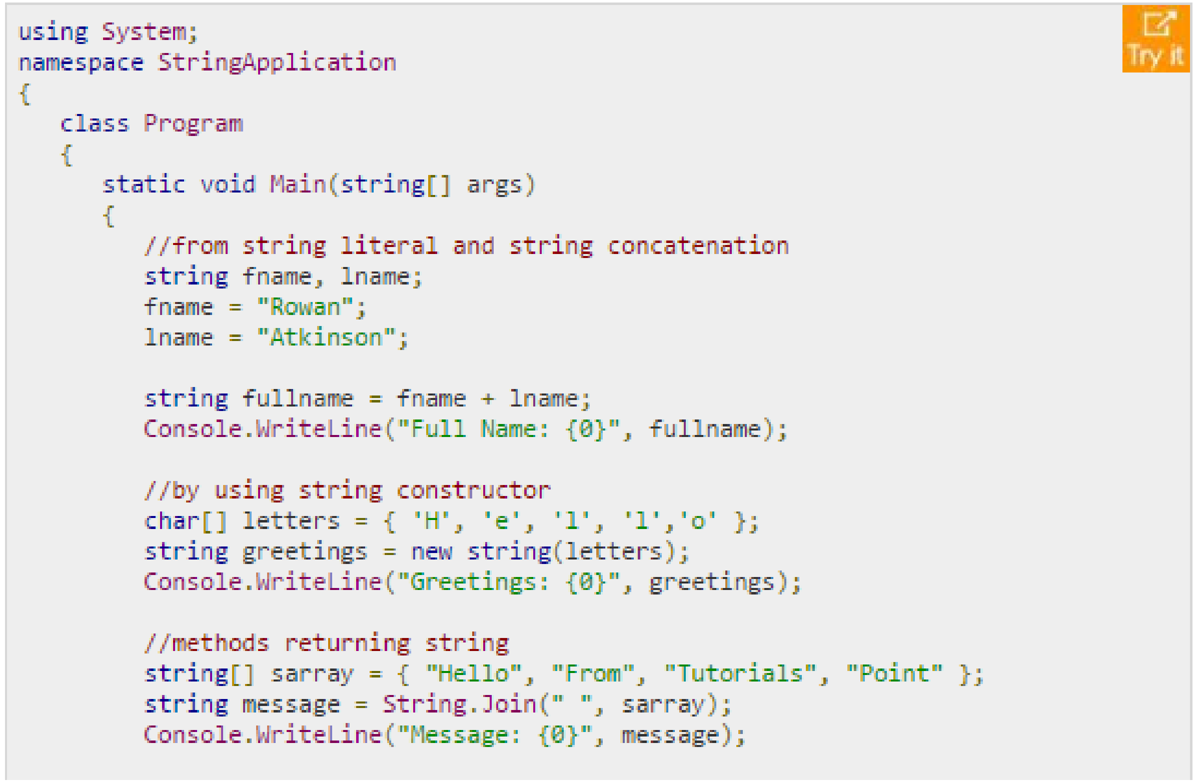
public string ToLower()

public string ToUpper()

public string Trim()

public char[] ToCharArray()

Strings (Some Other Methods with Example)



```
using System;
namespace StringApplication
{
   class Program
   {
      static void Main(string[] args)
      {
         //from string literal and string concatenation
         string fname, lname;
         fname = "Rowan";
         lname = "Atkinson";

         string fullname = fname + lname;
         Console.WriteLine("Full Name: {0}", fullname);

         //by using string constructor
         char[] letters = { 'H', 'e', 'l', 'l','o' };
         string greetings = new string(letters);
         Console.WriteLine("Greetings: {0}", greetings);

         //methods returning string
         string[] sarray = { "Hello", "From", "Tutorials", "Point" };
         string message = String.Join(" ", sarray);
         Console.WriteLine("Message: {0}", message);
```

String in C#

Running Demo