Republic of the Philippines
CEBU TECHNOLOGICAL UNIVERSITY
MAIN CAMPUS
M.J. Cuenco Ave. Cor. R. Palma Street, Cebu City, Philippines
Website: http://www.ctu.edu.ph

COLLEGE OF COMPUTER, INFORMATION AND COMMUNICATIONS TECHNOLOGY

ASP.NET

Module 7

C# String Methods

I. Preparation

At the end of this module students will:

- explain the methods of string;
- implement the string methods in simple problem;

II. Presentation

In C#, a string is a sequence of characters. For example, "hello" is a string containing a sequence of characters 'h', 'e', 'l', 'l', and 'o'.

We use the string keyword to create a string. For example,

// create a string

string str = "C# Programming";

String Length

A string in C# is actually an object, which contain properties and methods that can perform certain operations on strings. For example, the length of a string can be found with the Length property:

Example:

```
string str = "This is a string";
lblstrlength.Text = "The length of the string is: " + str.Length.ToString();
```

Output:

```
The length of the string is: 16
```

ToUpper() and ToLower()

- returns a copy of the string converted to uppercase or lowercase, example:

```
string str = "This is a string";
lblUp.Text = str.ToUpper();
lblLo.Text = str.ToLower();
```

Output:

```
Uppercase: THIS IS A STRING
Lowercase: this is a string
```

string.Concat()

- method to concatenate two strings, example:

```
string s1 = "Cebu";
string s2 = " Technological";
string s3 = " University";
lblUni.Text = string.Concat(s1, s2, s3);
```

Output:

```
Cebu Technological University
```

Access Strings

You can access the characters in a string by referring to its index number inside square brackets [ ]. This example prints the first character in str2:

Example:

```
string str2 = "Hello World";
for (int i = 0; i < str2.Length; i++) {
        Response.Write("index ["+i+"]: "+str2[i] + "<br/>");
    }
```

Output:

```
index [0]: H
index [1]: e
index [2]: l
index [3]: l
index [4]: o
index [5]:
index [6]: W
index [7]: o
index [8]: r
index [9]: l
index [10]: d
```

Indexof() method

The String IndexOf() method returns the index of the first occurrence of the specified character/substring within the string.

Syntax:

String.IndexOf(string value, int startindex, int count)

Example 1: C# String IndexOf()

```
string str2 = "Hello World";
lblIndx1.Text = str2.IndexOf("W").ToString();
```

Output:

6

Example 2: IndexOf() With Start Index

```
string str2 = "Hello World";
lblIndx1.Text = str2.IndexOf("W",0).ToString();
```

Output:

6

Example 3: IndexOf() With Start Index And Count

```
string str2 = "Hello World";
lblIndx1.Text = str2.IndexOf("W",0,5).ToString();
```

Output:

-1

Substring() method

The Substring() method returns a substring from the given string.

Syntax:

Substring(int startIndex, int length)

where:

startIndex - the beginning index of the substring
length - (optional) - length of the substring

Example 1:

```
string str2 = "Hello World";
```

```
lblIndx2.Text = str2.Substring(1);
```

Output:

```
ello World
```

Example 2:

```
string str2 = "Hello World. ASP.NET Students";
lblIndx2.Text = str2.Substring(0,9);
```

Output:

```
Hello Wor
```

Example 3:

```
string name = "John Wick, IV";
int pos = name.Index("W");
lblIndx2.Text = name.Substring(pos);
```

Output:

String Equals()

- method checks whether two strings have the same value or not.

Syntax:

String.Equals(string a, string b)

where:

a - first string to compare
b - second string to compare

The **Equals()** method returns:

True - if the strings are equal
False - if the strings are not equal

Example:

```
string str1 = "Hello World";
string str2 = "hello World";
if (String.Equals(str, str2))
     {
         Response.Write("String are equal");
     }
     else
```

```
{
    Response.Write("String are not equal");
}
```

Output:

```
String are not equal
```

Replace() method

- returns a new string by replacing each matching character/substring in the string with the new character/substring.

Syntax:

Replace(string oldValue, string newValue)

where:

oldValue - the substring to replace
newValue - new substring which will replace the old substring

Example:

```
string str1 = "Hollywood";
Response.Write(str1.Replace('H','B');

string str1 = "C# Programming";
Response.Write(str1.Replace("Programming","Copy and Paste");
```

Output:

```
Bollywood
C# Copy and Paste
```

String Remove()

- method removes a specified number of characters from the string.

Syntax:

Remove(int startIndex, int count)

where:

startIndex - index to begin deleting characters
count (optional) - number of characters to delete

Example:

```
string str1 = "Hollywood";
Response.Write(str1.Remove(5));

string str1 = "C# Programming";
Response.Write(str1.Remove(4,3);
```

Output:

```
Holly
C# Pramming
```

III. Performance Task

1. F.L.A.M.E.S is a game usually played among teens to find if they are compatible with their crushes. This game can determine the relationship between two people with the use of their names. Input the names of the two people and display the message:

   F = riends

   L = overs

   A = dmirers

   M = arriage

   E = nemies

   S = ecret Lovers