CS PART I - Unit 1 – Exercise 07 – More On Strings

Mr. Bellavia

EXERCISE 07 – MORE ON STRINGS

IMPORTANT: Before submission, make a copy of your **'Program.cs'** file for each question and then rename each file to the following:

File Names:

- *last_name_first name*_U1_E07_1.cs
- *last_name_first name*_U1_E07_2.cs
- *last_name_first name*_U1_E07_3.cs

*last_name_first name*_U1_E07_4.cs
*last_name_first name*_U1_E07_5.cs

Note: Along with last name and first name, make sure the end of the filename (i.e., before the **.cs**) has the **unit number**, **exercise number**, and **question number.** For example:

smith_*john*_***U1***_***E03***_***2***.*cs*

ATTENTION: Now that our programs are becoming more in depth, you will realize that there are many ways to code these programs!

1. Create the following string:

```
string myString = "Computer Science is the best course ever!";
```

Do the following:

a) Output each word to the console on a separate line by making use of the built-in **.Substring()** string function.

b) Output the **index** position of where the word **'course'** starts by making use of the built-in **.IndexOf()** string function.

2. Create the following string:

```
string s = "Led Zeppelin could be considered the greatest band of all time";
```

Now ask the user for an **index** (i.e., an **int**). Output the character at the given index position from string 's' above. **Hint:** Use the integer index inputted from the user inside square brackets on the string 's'.

3. Create a program that asks the user for their full name (assume the user simply types a first name with a space followed by a last name). Do the following:

a) Output how many characters their name is.

b) Store their first name and last name into separate variables and output each to the screen.

c) Store their initials into separate variables and output each to the screen.

HINT: for (b) and (c) you will need to make use of the built-in **.IndexOf()** string function.

4. The **.Remove()** string function can erase a piece of a string. For example:

```
string s = "";
s = "Computer Science is the best";
s = s.Remove(16, 7);
Console.WriteLine(s);
```

Output will be:

```
Computer Science best
```

The **.Remove()** string function has two parameters. The first is the **index** of where you want to start and the second is the **length** to remove. So, in the above example we are starting at index 16 and erasing 7 characters from that point.

Do not use **.Substring()** for these questions:

a) Write a program that asks the user to type a sentence. Then ask the user for the **index** they wish to start and the **length** they wish to erase. Output the resulting string.

b) Modify the above program so that instead of asking for the index and length, you ask for a **start index** and **end index.** You will need to modify your use of the **.Remove()** function to get this to work. **Hint:** You now have a start index and end index which you can use to calculate the appropriate length to erase.

5. Write a program that will ask the user to type a sentence. Then ask the user to type a word from that sentence that they wish to replace. Finally ask the user for a word that they would like to replace the previous word with. Output the resulting sentence.

Your **Input** **& Output** should work **exactly** like the following:

```
Enter a sentence: Hello Computer Science students in this class!

Enter a word from this sentence that you would like to replace: students

Enter a word to replace ‘students’ with: pupils

Hello Computer Science pupils in this class!
```

Note: There are many ways to do this! You may want to search on google other string functions such as **.Replace()** and/or **.Insert().**

BONUS: Modify the above program so that it is **NOT** case sensitive. **Hint:** Lookup **.ToUpper()** or **.ToLower()** for C# strings.