# Chapter 7 – String Handling

## Strings and string handling in programming

* A **string** is a collection of characters and is treated in C# as an object. A string is essentially an **array of characters**.
* A **character** is anything on a computer keyboard and is an elementary data type.

**It is frequently necessary as part of standard computer processing to look at STRINGS.**

C# allocates 2 bytes per character using Unicode – a coding system designed to cover all languages. The first 256 characters of Unicode are the same as ASCII:

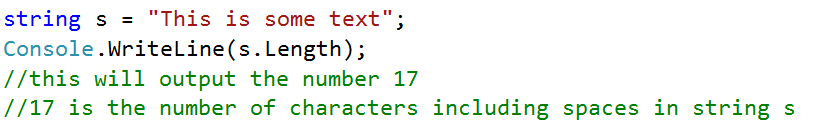
* Upper-case letters are stored as numbers 65–90
* Lower-case letters are stored as numbers 97–122
* Numeric digits 0 to 9 are stored as numbers 48–57
* A space character is stored as number 32

## Common operations you might perform on strings

What processes might involve looking at the individual characters within a string?

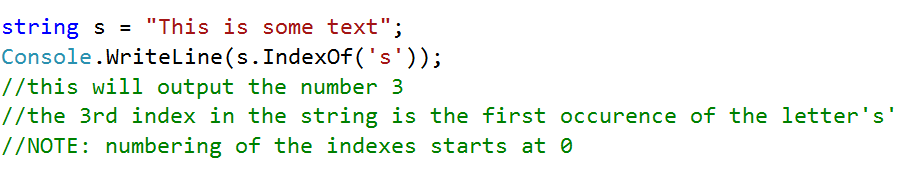
* Finding the **length** of the string (**string.Length**)

It can be useful to know the length of a string in case you want to loop through all its character elements, in which case you can use the string length to set the **for-loop** condition.

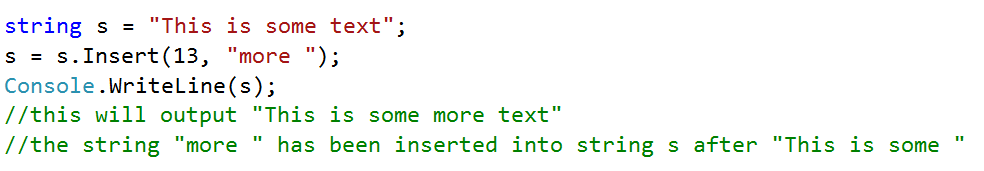


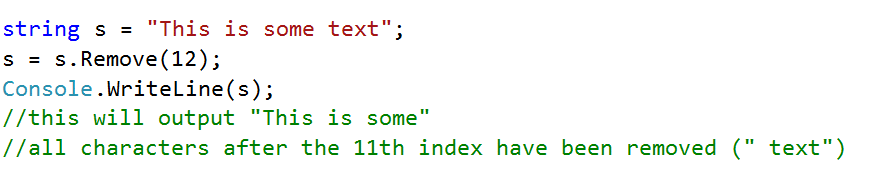
* Finding the **index** of the first occurrence of a value within a string (**string.IndexOf(“value”)**)

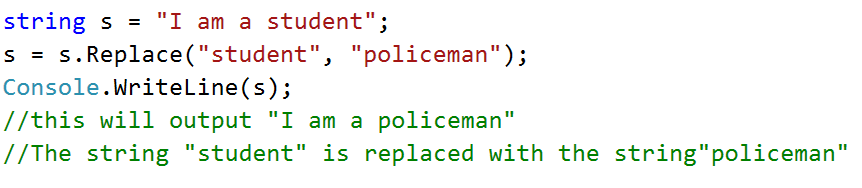
This is essentially finding out at what index point in the string a specified character first occurs. This string function will return a numerical value that is the index position number of the sought-after character in the string.



* **Inserting a value** into a string at a particular point (**string.Insert(Index, “value”)**)

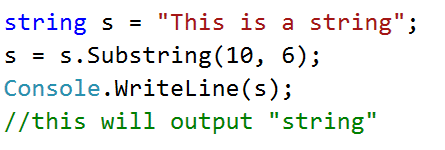


* **Removing characters** from a specified index onwards (**string.Remove(index)**)
* **Replace all instances of a specified value** in a string (**string.Replace(“oldvalue”, “newvalue”)**)



* **Return a substring of characters** starting at a particular **index** (**string.Substring(index, length)**)

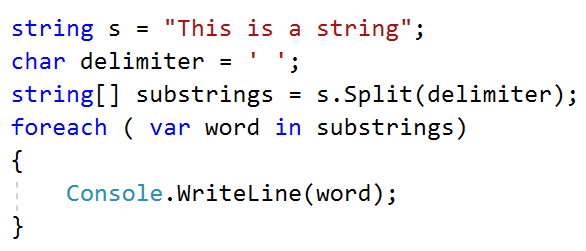
The first parameter is the starting position of the new string. The second parameter refers to the number of characters that will be extracted.



* **Returns a true or false value** if a string contains a particular value (**string.Contains(“value”)**)



* **Splits a string** (**string.Split(‘character’)**)



In this example the string is being split where there is a space (an empty character).

Research Task: *Other String Methods in C#*

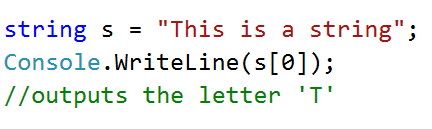
The examples above show some of the most common and useful string-handling methods.

Do some research to find out about **at least two** other methods that can be used on strings.

Make a note of what they do and what syntax is required.

## Accessing individual characters in a string

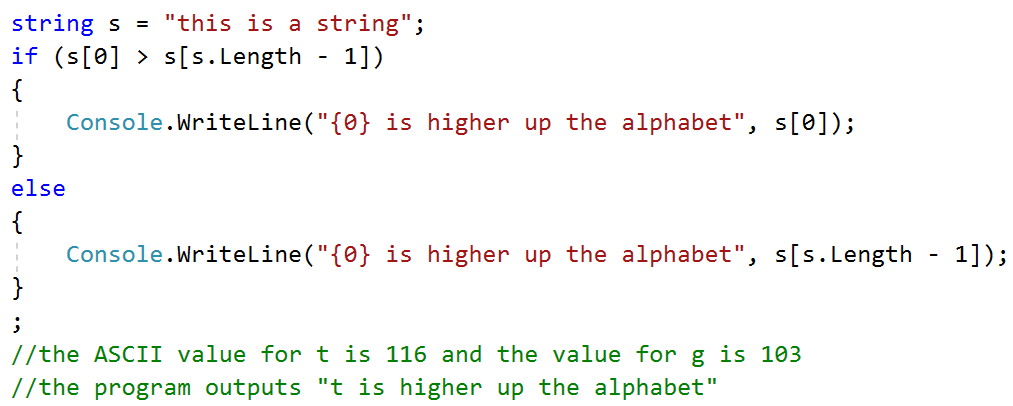
You can also access characters in a string by its index – just like you would when accessing elements of an array:



## Processing strings with relational operators

It is possible to use **relational operators** with strings. These operators work on the ASCII value / Unicode number.

In the example below, the value of the lower-case letter ‘t’ is 116 and the value of the lower-case letter ‘g’ is 103. It is these ASCII values that are compared in the selection condition.

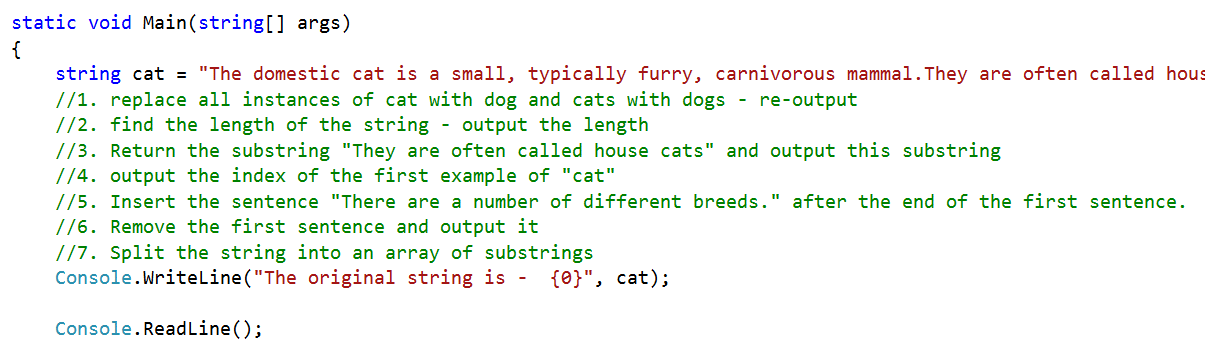




**{ }**

Coding Task: *Program 7.1 – The Cat String Program*

Open the **Program 7.1.txt file**, copy it into a new console mode application and follow the instructions to manipulate the string and output amended strings.





**{ }**

Coding Task: *Program 7.2 – The Space Wars Name Program*

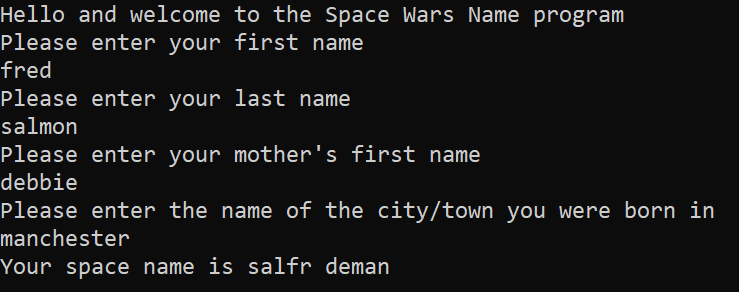
You are to write a program that comes up with your space wars name by extracting characters from   
input values and concatenating them.

**For your first spacename you need to:**

* take the first three letters of your real last name
* add the first two letters of your real first name

**For your last spacename you need to:**

* take the first two letters of your mother’s first name
* add the first three letters of the city in which you were born

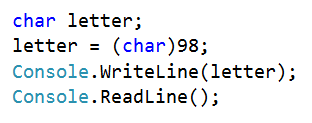
The screenshot below shows typical output to the console window.

## Working with strings using ASCII values

There are a number of different operations using the ASCII values of characters that are often useful when programming with strings.

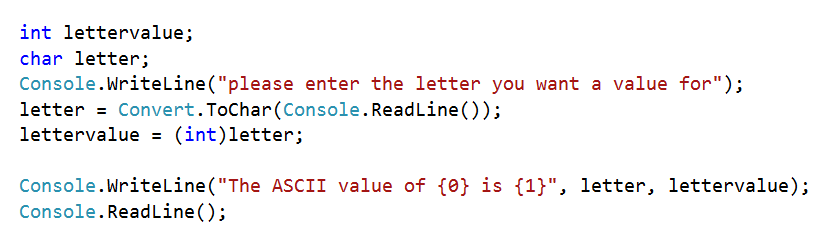
## Assigning a character based on the ASCII value

Here the variable **letter** is being assigned the character ‘**b**’ using its ASCII value.



## Finding the ASCII value of a character

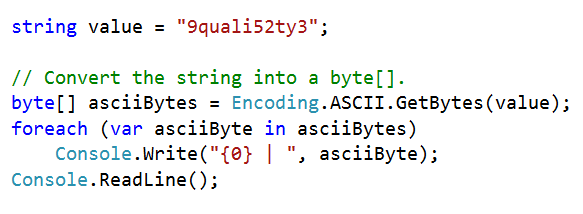
This program takes in a character, finds its ASCII value and then outputs that value.





## Finding the ASCII values in a string

This program takes in a string, works out the ASCII value of each character and stores them in an array.







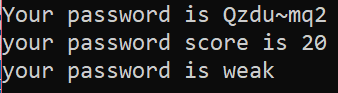
**{ }**

Coding Task: *Program 7.3 – Password Generator*

Write a password generator program that outputs an **eight-digit** password using the following steps:

* 1. Generate a **random number** (between 33 and 127)
  2. Use that number to generate a **character**
  3. Add that character to a **password string**
  4. Check the password string to see whether there is a **capital letter** – if there is, add 10 to the point score for each letter (ASCII values for capitals are 65 to 90)
  5. Check the password string for any of the **non-numeric or non-alphabetical characters** – add 5 for each one found
  6. **Output the password and its strength** – if the score is 40 or more, then it is high strength; if it is 25 or more, it is medium strength; if it is < 25 then it is weak.

The image below shows a typical output.



## Chapter 7 – Consolidation Tasks

Program 7.6

You are to complete a string-checking program in a C# console mode application.

Complete a program for string checking that will:

* Feed in a string containing the sentence ‘I am going to check every word of this sentence for the keywords’
* Split the string into an array of the individual words making up the sentence
* Check the array of words for instances of the keywords (check, word, sentence)
* Output the sentence with the keywords highlighted (in a different console colour)

The image below shows typical expected output.



**Evidence required:**

1. An **annotated code listing** for the above program.

1. A **testing screenshot** showing the required output.

### TIP! Changing the colour of the console window and console text

This statement allows you to change the **console window background colour**:



This statement allows you to change the colour of the **text**:



## Activities

Research question: