

2. StringBuilder

StringBuilder

- Defined in System.Text
- A mutable string
- Easy and fast to create and manipulate strings

Not for Searching

- `IndexOf()`
- `LastIndexOf()`
- `Contains()`
- `StartsWith()`
- ...

String Manipulation Methods

- Append()
- Insert()
- Remove()
- Replace()
- Clear()

```
using System;
using System.Text;

namespace StringManipulate
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder();
            builder.Append('-', 10);
            Console.WriteLine(builder);
        }
    }
}
```



```
using System;
using System.Text;

namespace StringManipulate
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder();
            builder.Append('-', 10);
            builder.AppendLine();
            builder.Append("Header");
            builder.AppendLine();
            builder.Append('-', 10);
            Console.WriteLine(builder);
        }
    }
}
```

> Terminal – StringManipulate

Header

█

```
using System;
using System.Text;

namespace StringManipulate
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder();
            builder.Append('-', 10);
            builder.AppendLine();
            builder.Append("Header");
            builder.AppendLine();
            builder.Append('-', 10);
            builder.Replace('-', '+');
            Console.WriteLine(builder);
        }
    }
}
```

>- Terminal – StringManipulate

```
+++++++
Header
+++++++
```

```
using System;
using System.Text;

namespace StringManipulate
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder();
            builder.Append('-', 10);
            builder.AppendLine();
            builder.Append("Header");
            builder.AppendLine();
            builder.Append('-', 10);
            builder.Replace('-', '+');
            builder.Remove(0, 10);
            Console.WriteLine(builder);
        }
    }
}
```

Terminal – StringManipulate

```
Header
+++++
```



```
using System;
using System.Text;

namespace StringManipulate
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder();
            builder.Append('-', 10);
            builder.AppendLine();
            builder.Append("Header");
            builder.AppendLine();
            builder.Append('-', 10);
            builder.Replace('-', '+');
            builder.Remove(0, 10);
            builder.Insert(0, new string('-', 10));
            Console.WriteLine(builder);
        }
    }
}
```

Terminal – StringManipulate

```
-----
Header
+++++++

```

```
using System;
using System.Text;

namespace StringManipulate
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder();
            builder.Append('-', 10);
            builder.AppendLine();
            builder.Append("Header");
            builder.AppendLine();
            builder.Append('-', 10);
            builder.Replace('-', '+');
            builder.Remove(0, 10);
            builder.Insert(0, new string('-', 10));
            Console.WriteLine(builder);

            Console.WriteLine("First Character: {0}", builder[0]);
        }
    }
}
```

Terminal – StringManipulate

```
-----
Header
+++++++
First Character: -
█
```

```
using System;
using System.Text;

namespace StringManipulateUpdated
{
    class Program
    {
        static void Main(string[] args)
        {
            var builder = new StringBuilder();
            builder
                .Append('-', 10)
                .AppendLine()
                .Append("Header")
                .AppendLine()
                .Append('-', 10)
                .Replace('-', '+')
                .Remove(0, 10)
                .Insert(0, new string('-', 10));
            Console.WriteLine(builder);
        }
    }
}
```

Terminal – StringManipulateUpdated

```
-----
Header
+++++
```

Exercises

Note: For all these exercises, ignore input validation unless otherwise specified. Assume the user provides input in the format that the program expects.

1- Write a program and ask the user to enter a few numbers separated by a hyphen. Work out if the numbers are consecutive. For example, if the input is "5-6-7-8-9" or "20-19-18-17-16", display a message: "Consecutive"; otherwise, display "Not Consecutive".

2- Write a program and ask the user to enter a few numbers separated by a hyphen. If the user simply presses Enter, without supplying an input, exit immediately; otherwise, check to see if there are duplicates. If so, display "Duplicate" on the console.

3- Write a program and ask the user to enter a time value in the 24-hour time format (e.g. 19:00). A valid time should be between 00:00 and 23:59. If the time is valid, display "Ok"; otherwise, display "Invalid Time". If the user doesn't provide any values, consider it as invalid time.

4- Write a program and ask the user to enter a few words separated by a space. Use the words to create a variable name with PascalCase. For example, if the user types: "number of students", display "NumberOfStudents". Make sure that the program is not dependent on the input. So, if the user types "NUMBER OF STUDENTS", the program should still display "NumberOfStudents".

5- Write a program and ask the user to enter an English word. Count the number of vowels (a, e, o, u, i) in the word. So, if the user enters "inadequate", the program should display 6 on the console.